Timm Grams

Grundlagen des Qualitäts- und Risikomanagements

Zuverlässigkeit, Sicherheit, Bedienbarkeit

Q&R-v5
Ausgabe vom 18.04.2008

Vorwort

Die *Objektivität* der wissenschaftlichen Sätze liegt darin,
dass sie *intersubjektiv nachprüfbar* sein müssen.
Karl R. Popper, 1982

Für mich besteht kein Kontrast zwischen "persönlich" und "sachlich".
Bassam Tibi, 1998

Wenn man ganz präzise erzählt, kann man vielleicht universell werden,
wenn man versucht, universell zu sein, kommt nur fades Zeug raus.
Doris Dörrie in DER SPIEGEL 11/1994

Der Rahmen ist weit gesteckt: Es geht um das nutzbringende und gefahrlose Zusammenspiel von Mensch, Maschine und Umwelt. Bei einem so allgemeinen Thema wie dem Qualitäts- und Risikomanagement droht stets die Gefahr, dass die Rede im Allgemeinen bleibt, dass nur ein weiteres Mal ein Lebenszyklusmodell entsteht, und dass allgemeine Organisationspläne, Verhaltensvorschriften und Checklisten neu sortiert zu Papier gebracht werden. Die Relevanz des Dargebrachten bleibt im Dunkeln, weil die Unmittelbarkeit des Selbsterlebten nicht gegeben ist, und weil die Verbindungen zu den Erfahrungen - auch denen anderer - fehlen.

Die Eingangszitate will ich als Leitschnur nehmen, um dem zu entgehen. Nicht ums Allgemeine geht es mir, sondern um das Verallgemeinerbare. Ich strebe nach einer persönlichen, konkreten, knappen und möglichst objektiven Darstellung des Stoffes.

Das Persönliche: Allein die Stoffauswahl, die jeder Autor treffen muss, macht ein Buch zu einer sehr persönlichen Angelegenheit. Dennoch herrscht in der wissenschaftlichen Literatur ein unpersönlicher Stil vor. Dem folge ich nicht, denn ein solcher Stil bietet nur den Anschein der Objektivität. Er trägt nicht von selbst zur Klarheit bei, sondern eher dazu, Verantwortung zu verbergen.

Ich versuche das, was mir im Laufe meiner Industrie- und Lehrtätigkeit sowie in meiner Arbeit in verschiedenen VDI/VDE-Arbeitskreisen für die industrielle Praxis als wichtig erschienen ist, möglichst knapp und lückenlos darzustellen. Die dargebotenen Grundlagen sollen Orientierung ermöglichen und die Basis für ein tieferes Eindringen in Spezialfragen bieten.

Am Anfang steht ein konsistentes Begriffssystem. Dabei lege ich ein äußerst einfaches Modell des Mensch-Maschine-Umwelt-Systems zu Grunde, in dem der Begriff der *Spezifikation* Dreh- und Angelpunkt ist. Es bietet eine tragfähige Basis für ein in sich widerspruchsfreies Begriffssystem. Dieser Ansatz wurde beispielsweise in die Richtlinie VDI/VDE 3542/4 übernommen.

Maschinen lassen sich als Realisierungen von Spezifikationen auffassen. Zur Beschreibung von Maschinen gehen wir also von einem hohen Abstraktionsniveau aus. Dieses Vorgehen ist *top down.* Sobald die Komponente Mensch und deren Verhalten zum Gegenstand der Betrachtung wird, ist es günstiger, vom Konkreten auszugehen - also *bottom up.* Der unterschiedliche Blickwinkel liegt in der Natur der Sache: Maschinen machen wir, Natur und Menschen finden wir vor.

Eine weiteres Anliegen ist, die wichtigsten relevanten Ergebnisse nichttechnischer Disziplinen wie Ökonomie, Verhaltensforschung, Psychologie, Soziologie und Philosophie den Grundlagen des Faches Risikomanagement zuzurechnen.

Das Konkrete: Eine umfassende Darstellung der Methoden aller betroffenen Wissensgebiete ist in einem interdisziplinär angelegten Buch kaum machbar. Das eklektische Zusammentragen von zwar nützlichem aber zusammenhanglosem Wissen führt andererseits auch nicht weit. Unser Denkapparat könnte mit einer ungeordneten Sammlung kaum etwas Vernünftiges anfangen. Als Ordnungsprinzip und Gliederungsschema bietet sich der *Lernregelkreis* an. Er besteht aus dem Dreischritt (1) konkrete Fehler aufzeigen, (2) deren Ursachen analysieren, und (3) Lehren daraus ziehen und in Form von Regeln verallgemeinern.

Das *Lernen aus den Fehlern* ist als Prinzip von überragender Bedeutung. Es wird in den angelsächsischen Länder nach meiner Beobachtung stärker gepflegt als bei uns. Die Challenger-Katastrophe beispielsweise wurde von einer unabhängigen Kommission untersucht. Die Ergebnisse stehen der Öffentlichkeit zur Verfügung. Es gibt mehrere tiefgründige und teils kontroverse Analysen (Feynman, Vaughan, Perrow). Das typische Lehrbuch zum Themenkreis Mensch-Maschine-Sicherheit enthält, wenn es aus den USA oder England kommt, Unfallberichte und Lehren daraus (Leveson, Reason).

In deutschen Lehrbüchern zum Thema konnte ich wenig Vergleichbares finden. Beispielsweise frage ich mich, warum in einem ansonsten hervorragenden Buch zum Thema Qualitätsmanagement als Beispiel für einen Unglücksfall ausgerechnet die griechische Sage des Ikarus herhalten muss. Demgegenüber stellen die Berichte von Unfällen der jüngeren Vergangenheit einen teuer erworbenen Wissensschatz dar. Aus ihnen lassen sich lebenswichtige Lehren ziehen.

Das Sachlich-Objektive: Voraussetzung für eine ordentliche Ingenieursarbeit ist die Beherrschung der mathematischen und naturwissenschaftlichen Grundlagen. Dabei geht es nicht nur um Formeln, sondern auch um deren Gültigkeits- und Anwendungsbereiche. Nicht allein die richtige Formel macht's, sondern die damit verbundene präzise, unmissverständliche und nachprüfbare Aussage.

Wirklich schwierig wird es, wenn der Mensch selbst Betrachtungsobjekt ist. Hier ist Objektivität - die *Kritisierbarkeit* und intersubjektive Nachprüfbarkeit der Sachargumente - noch viel schwerer zu bekommen als in den naturwissenschaftlich-technischen Disziplinen. Dennoch: Der Standpunkt - sei er auch persönlich - wird klar und unmissverständlich mitgeteilt. Die Beweggründe bleiben nachvollziehbar. Andere Standpunkte und Deutungen werden dadurch nicht ausgeschlossen.

Ich fasse zusammen: Beim Qualitäts- und Risikomanagement geht es um die Bewertung von Objekten und die daraus folgenden Entscheidungen, aber immer auch um die wertenden und entscheidenden Personen. Die Objektivität liegt darin, dass uns beide Seiten der Medaille bewusst sind - und dass wir beide Seiten zum Gegenstand der kritischen Würdigung machen.

Fulda, 21. April 2001 Timm Grams

Änderungsbericht

Bis Q&R-v4: Nur kleine redaktionelle Veränderungen gegenüber dem Vieweg-Buch

Q&R-v5, 18.04.2008: Bayes-Schätzung (4.3) hinzugefügt mit entsprechenden Anpassungen

Inhaltsverzeichnis

Einleitung

"Qualität ist, wenn der Mensch zurückkommt und nicht das Produkt"
Stefan Schmitz, Berlin 1993 ("100 Jahre VDE")

Jeder kennt solche Geschichten aus seinem Alltag: Ein neuer Diaprojektor muss her. Der alte hat ausgedient, weil es inzwischen ein neues System von Diarähmchen gibt (1). Die Dias sind nun schön platzsparend, werden aber vom alten Gerät nicht mehr verkraftet.

Zuhause wird ausgepackt und gleich ausprobiert. Auf dem Bedienungsteil gibt es *einen* Knopf mit zwei Pfeilen (2). Der ist für den Diawechsel. Drückt man auf den Pfeil nach vorne, kommt das nächste Dia. Drückt man auf den Pfeil nach hinten, geht es nicht etwa zurück, sondern es geschieht dasselbe wie vorher. Daraufhin wird der Laden aufgesucht und eine Beschwerde vorgebracht. Der Verkäufer lächelt nachsichtig: "Nächstes Dia: kurz drücken. Zurück zum vorhergehenden: länger drücken". Beschämt zieht man von dannen.

Eine Weile tut das Gerät seinen Dienst. Als erstes - und zwar schon nach nur wenigen Diavorführungen - entledigt sich der Multifunktionsknopf eines Teils seiner Funktionen (3): Ab nun geht es nur noch vorwärts. Im Laufe der Jahre geht hin und wieder eine Birne kaputt (4). Das ist nicht weiter schlimm. Da man damit rechnen muss, ist stets eine Ersatzbirne (5) im Haus. Nach ein paar Diaabenden bricht die Einstellschraube eines Fußes ab (6). Ab jetzt müssen Atlanten oder Kochbücher zum Justieren des Apparats herhalten (7). Bald darauf geht auch noch einer der Diaschieber kaputt. Die Optik zeigt auch bereits Eintrübungen (8). Das Gerät ist Schrott.

Nicht immer kommt in kurzer Zeit so viel zusammen. Aber grundsätzlich muss man bei jedem Gerät damit rechnen, dass man nicht alles kriegt, was man möchte (1), dass es zu Bedienfehlern (2) und zu Teil- (3, 6) oder Totalausfällen (4) kommt. Diese werden toleriert, repariert (5) oder auch durch Notmaßnahmen (7) überbrückt - solange, bis das erträgliche Maß überschritten ist (8).

Qualität zeigt sich darin, dass die *gewünschten* Funktionen *dauerhaft* erbracht werden. Dass die richtigen Funktionen erbracht werden, kann man bei einfachen Geräten durch Ausprobieren feststellen. Bei komplexen Dingen reicht das nicht, und wir müssen den Herstellerangaben trauen. Noch problematischer wird es, wenn wir etwas über die Haltbarkeit eines Produkts wissen wollen: Es geht um das Vorhersagen des zukünftigen Verhaltens. Das scheint eher eine Sache für Wahrsager zu sein, und nicht die von Ingenieuren.

Dennoch: Es gibt eine seriöse technische Wissenschaft, die sich mit solchen Vorhersagen befasst. Es ist die Lehre von der *Zuverlässigkeit und Sicherheit*. Sie hat sich etwa seit dem Zweiten Weltkrieg zu einer achtbaren und äußerst erfolgreichen Disziplin entwickelt. Sie ist wesentlicher Bestandteil der *Qualitätssicherung* und des *Risikomanagements* technischer Einrichtungen und Produkte.

Das Buch stellt die Grundlagen der Zuverlässigkeit, Sicherheit und Bedienbarkeit technischer Objekte dar. Es geht um die Beantwortung der Fragen: Was genau ist unter Zuverlässigkeit und Sicherheit zu verstehen? Wie lassen sich Zuverlässigkeit und Sicherheit messen? Mit welchen - von der speziellen Technologie unabhängigen - Mitteln lassen sich vorgegebene Zuverlässigkeitsziele erreichen? Welche Methoden der Zuverlässigkeitsprognose und des Zuverlässigkeitsnachweises gibt es, und wie werden sie angewandt? Wie treffsicher sind die Aussagen? Welche Rolle spielen die Wartungs- und Reparaturstrategien? Auf welche besonderen Fähigkeiten und Schwächen des Menschen ist Rücksicht zu nehmen? Welchen Platz haben Zuverläs-

sigkeits- und Sicherheitstechnik im Rahmen eines umfassenden Qualitäts- und Risikomanagements?

Noch eine Anmerkung zum angesprochenen Multifunktionsknopf: Dass sich die Sache mit dem Multifunktionsknopf heute durchgesetzt hat, spricht nicht für dieses Design. Es spricht eher gegen uns Verbraucher: Wir sind nicht selbstbewusst genug, um uns gegen solchen Unfug zu wehren. Wir neigen - auch aus vermeintlicher Bequemlichkeit - dazu, uns mit unpraktischem Schnickschnack zu arrangieren. Dass es sich hier tatsächlich um einen Designfehler handelt, hat Donald Norman in seinem Buch "The Design of Everyday Things" unterhaltsam dargestellt.

1 Grundbegriffe

1.1 Fehler im Mensch-Maschine-Umwelt-System

Der Werkzeuggebrauch hat uns Menschen zu einer überaus erfolgreichen Art werden lassen. Sogar das Schicksal des gesamten Planeten hängt heute davon ab, wie wir mit der Technik umgehen. Von Anfang an schlagen wir uns auch mit den negativen Seiten des Gebrauchs von Werkzeugen und Maschinen herum.

Unter einer Maschine wollen wir verstehen: ein Automobil, eine Lokomotive, einen Aufzug, ein Kraftwerk, ein Haushaltsgerät, ein Telefon, einen Personal Computer, eine Werkzeugmaschine, eine verfahrenstechnische Anlage, eine automatisierte Produktionslinie, oder manches andere. Für die Maschine möge es eine klare Beschreibung dessen geben, was sie tun soll. Das ist ihre Spezifikation.

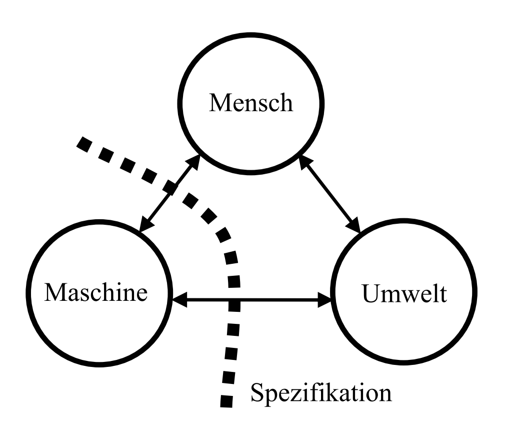


Bild 1.1 Das Mensch-Maschine-Umwelt-System

Im Mensch-Maschine-Umwelt-System (Bild 1.1) steht die „Komponente Mensch“ für den Bediener oder Anwender und nicht für den Hersteller der Maschine.

Das Zusammenspiel von Mensch, Maschine und Umwelt führt immer wieder zu unerwünschten Resultaten. Das kann daran liegen, dass die Maschine

- schlecht gebaut worden ist,
- nicht auf Dauer das tut, was sie tun soll,
- nicht richtig bedient wird,
- böswillig missbraucht wird,
- Wirkungen entfaltet, die nicht vorhergesehen worden sind.

Von *Fehlern* sprechen wir nur dann, wenn die Ursachen der unerwünschten Resultate im Menschen oder in der Maschine liegen. Der Kasten "Fehlerklassifikation" zeigt in einer Übersicht diejenigen Fehlerklassen, die in diesem Buch eine Rolle spielen.

Fehlerklassifikation

- Fehler
  - Spezifikationsfehler
  - Technischer Fehler
    - Eingebauter Fehler (inhärenter Fehler)
      - Konstruktionsfehler
        - Programmierfehler
        - Entwurfsfehler in der Hardware
      - Fertigungsfehler
    - Ausfall (physikalischer Fehler)
  - Bedienfehler (fälschlich: „menschliches Versagen“)

Da unter Fehler eine *Abweichung von der Norm* zu verstehen ist, muss zunächst geklärt werden, welche Normen verbindlich sind oder allgemein anerkannt werden. Zu den normativen Vorgaben gehört die konkrete Spezifikation der Maschine.

1.2 Spezifikation und Zuverlässigkeit

Jede *Spezifikation* hat zwei Wirkungsrichtungen. Sie grenzt Verantwortungsbereiche voneinander ab. Erstens legt sie für Entwicklung, Konstruktion und Fertigung fest, unter welchen Bedingungen das System welche Leistungen zu erbringen hat, und zweitens sagt sie dem Bediener und Anwender, was er zu tun hat, um vom System diese Leistungen zu bekommen. Ob ein System wirklich nützlich und unschädlich ist, hängt nicht nur davon ab, ob es spezifikationsgemäß funktioniert, sondern auch davon, inwieweit es sich vom Menschen mit seinen angeborenen und erworbenen Fähigkeiten und Schwächen richtig bedienen lässt. Die Bewertung von Nutzen und Risiko muss neben den Fehlern des Systems auch die möglichen Bedienfehler - also die anwenderseitigen Verstöße gegen die Spezifikation - berücksichtigen. *Zuverlässigkeit* besagt nur, inwieweit ein System die Spezifikation erfüllt. (Diese Formulierung wurde - wie noch ein paar andere dieses Kapitels - in die VDI/VDE-Richtlinie 3542/4 übernommen.)

Die Spezifikation bildet den Ausgangspunkt aller Begriffsbestimmungen zum Thema Zuverlässigkeit und Sicherheit. Zur Präzisierung werden die wichtigsten Zusammenhänge mittels Mengen und Relationen dargestellt.

Zugrundegelegt wird das Blockschaltbild eines Systems (Maschine), Bild 1.2. Die Menge der Eingabedaten des Systems wird mit X und die der Ausgabedaten mit Y bezeichnet. Jedes Datum dieser Mengen beschreibt im Allgemeinen eine mehrdimensionale Zeitfunktion - also den Zeitverlauf einer zusammengesetzten Ein- bzw. einer Ausgangsgröße des Systems. Mit x und y bezeichnen wir Variablen mit Werten aus X bzw. Y und mit $x(t)$ bzw. $y(t)$ deren Momentanwerte zum Zeitpunkt t.

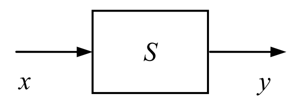


Bild 1.2 Blockschaltbild der Maschine (des Systems)

Die *Spezifikation* R des Systems möge als *Ein-Ausgaberelation* vorliegen: $R \subseteq X \times Y$. Jedes Wertepaar (x, y) aus R besagt, dass x ein zulässiger Eingabewert und y ein zugehöriger zulässiger Ausgabewert ist. Das Verhalten des konkret vorliegenden Systems wird durch die Ein-Ausgaberelation S beschrieben: $(x, y) \in S$ heißt, dass der mögliche Eingabewert x bei diesem System die Ausgabe y zur Folge haben kann.

Der Definitionsbereich einer Relation R wird hier mit dom(R) bezeichnet und der Wertebereich mit range(R):

- $\text{dom}(R) = \{x \mid \text{Es gibt ein } y, \text{ so dass } (x, y) \in R\}$
- $\text{range}(R) = \{y \mid \text{Es gibt ein } x, \text{ so dass } (x, y) \in R\}$

Der Einfachheit halber setzen wir voraus, dass das betrachtete System deterministisch ist. Bei *deterministischen Systemen* gibt es zu jedem zulässigen Eingabewert x genau einen Ausgabewert y. Dann handelt es sich bei S um eine *Funktion* im mathematischen Sinn und man schreibt $y = S(x)$ anstelle von $(x, y) \in S$.

Für deterministische Systeme gilt das *Korrektheitskriterium von Mills* (1986): Ein deterministisches System ist genau dann *korrekt* (funktionsfähig), wenn $\text{dom}(S \cap R) = \text{dom}(R)$, wenn also die Menge der korrekt behandelten Eingaben gleich dem Definitionsbereich der Spezifikation ist. Ein *technischer Fehler* liegt genau dann vor, wenn $\text{dom}(S \cap R) \neq \text{dom}(R)$ gilt, Bild 1.3.

Nehmen wir als Beispiel ein stark vereinfachtes Modell für einen

Stromkreis. Zwei Variablen für die Eingangsgrößen wollen wir unterscheiden: die Stromversorgung des Hauses und den Schalter für das Ein-/Ausschalten. Wenn wir die Werte der Größen als geordnete Paare (Stromversorgung, Schalter) darstellen, dann erhalten wir für die Menge der Eingabedaten X = {(ein, ein), (ein, aus), (aus, ein), (aus, aus)}. Die Ausgabevariable ist der Strom. Er fließt (ein) oder er fließt nicht (aus). Die Menge der Ausgabedaten ist Y = {ein, aus}. Mit diesen stark vergröberten Festlegungen erhält man die Spezifikation des Stromkreises zu R = {((ein, ein), ein), ((ein, aus), aus), ((aus, ein), aus), ((aus, aus), aus)}. Falls beim Eingabedatum (ein, ein) die Lampe aus bleibt, liegt ein technischer Fehler vor.

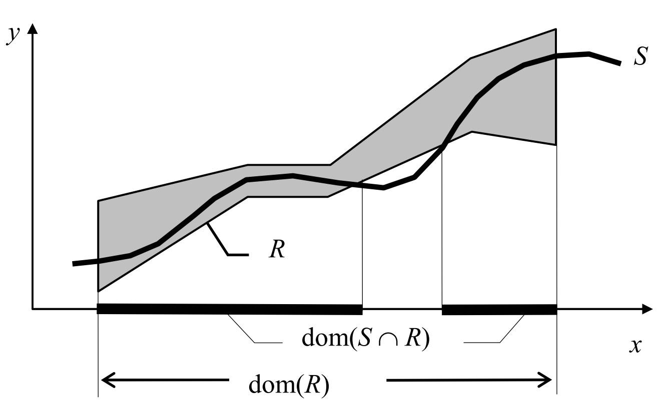


Bild 1.3 Spezifikation und Fehler

1.3 Sicherheitsspezifikation

Da in komplexen Systemen - also in Systemen aus vielen Komponenten, die miteinander eng verkoppelt sind - erfahrungsgemäß Fehler stecken oder mit der Zeit hineinkommen, muss man erreichen, dass von diesen Systemen auch dann, wenn Subsysteme fehlerhaft sind, keine *Gefahr* ausgeht.

Für Systeme in sicherheitsbezogenen Anwendungen (z.B. Steuerung von Anlagen mit großem Gefährdungspotential) wird das geforderte Verhalten in einer *Sicherheitsspezifikation* definiert. Die ursprüngliche Spezifikation wird zur Unterscheidung *funktionale Spezifikation* genannt.

In der Sicherheitsspezifikation wird vom System im Allgemeinen weniger verlangt als in der Spezifikation der Funktion: Das System muss nicht mehr unter allen Umständen voll funktionsfähig sein; es wird lediglich verlangt, dass von ihm keine Gefahr (im Sinne eines drohenden Schadens) für Mensch und Umwelt ausgeht.

Die Sicherheitsspezifikation wird zur Unterscheidung von der auf die normale Funktion bezogenen Spezifikation mit einem Strich gekennzeichnet. Für die Sicherheitsspezifikation schreibt man also R'.

Durch die Sicherheitsspezifikation wird die Menge der zulässigen Ein-Ausgabepaare in der Regel vergrößert. Wir stellen also die folgende Grundforderung an die Sicherheitsspezifikation: $R \subseteq R'$. Zur Erläuterung bleiben wir beim Beispiel

Stromkreis (Fortsetzung). Die Sicherheitsspezifikation für diesen Stromkreis verlangt, dass bei Kurzschluss und eingeschalteter Stromversorgung des Hauses kein gefährlicher Überstrom auftreten darf. Im Falle eines Kurzschlusses soll der Strom mittels einer Sicherung abgeschaltet werden. Da wir den Blick jetzt geweitet haben und die Möglichkeit eines Kurzschlusses in Betracht ziehen, erweitern wir die Eingangsgröße um eine entsprechende Komponente. Bild 1.4 zeigt die Spezifikationen und deren Beziehung zueinander. Die Sicherheitsspezifikation

ist Obermenge der funktionalen Spezifikation: In unserem Fall ist der Definitionsbereich größer und es gibt für einen Eingangswert einen größeren Spielraum der Ausgangswerte.

Wenn in einem komplexen System die Hardware ausfällt, dann wird das zunächst einmal dazu führen, dass die funktionale Spezifikation nicht mehr erfüllt ist. Bezüglich der Sicherheitsspezifikation kann das System aber nach wie vor korrekt sein. Ein *sicherheitsbezogener Ausfall* ist geschehen, wenn auch die Sicherheitsspezifikation verletzt ist.

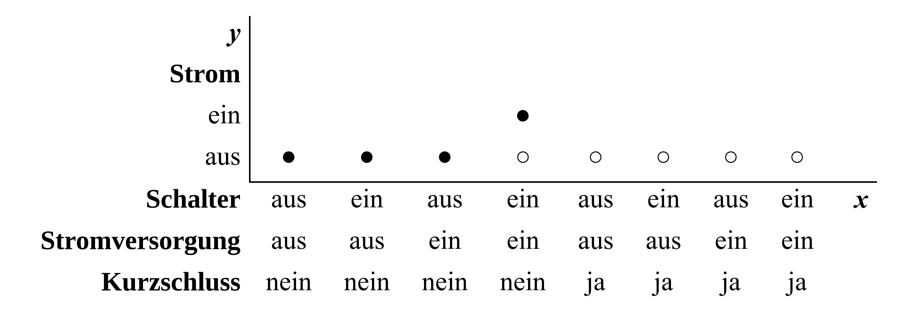


Bild 1.4 Funktionale Spezifikation (●) und Sicherheitsspezifikation (● und ○) eines Stromkreises

1.4 Spezifikation und Bedienfehler

Der Begriff "menschlicher Fehler" ist in Analogie zum technischen Fehler gebildet worden. Er bezeichnet jedoch nicht etwa Dinge wie Mordlust und Kleptomanie. Vergleichsweise profane Vorkommnisse wie das irrtümliche Betätigen eines Schalters, den man besser nicht angefasst hätte, werden als "menschliche Fehler" eingestuft. Man hat schon Computer gesehen, bei denen ein Reset ausgelöst werden kann, wenn man nur die Tastatur von sich wegschiebt. Soll sich der Anwender eines "menschlichen Fehlers" bezichtigen lassen, nur weil der Designer eine funktionswidrige Anordnung der Reset-Taste besonders schön fand?

Besser scheint zu sein, den Begriff "menschliches Versagen" durch *Bedienfehler* zu ersetzen und diesen von vornherein techniknah und unter Ausklammerung der Schuldfrage zu fassen (Grams, 1998).

Ein *Bedienfehler* ist definiert als ein bedienerseitiger Verstoß gegen die Spezifikation. Ein Bedienfehler liegt also genau dann vor, wenn eine getätigte oder auch unterlassene Bedienhandlung zu einer Eingabe in das System führt, die - gemessen an der Spezifikation und am angestrebten Ziel - unzulässig ist.

Erläuterung: Sei R die Spezifikation des Systems in Form einer Ein-Ausgaberelation. Sei ferner B das angestrebte Resultat: $B \subseteq \mathrm{range}(R)$. Mit A wird die zugehörige *Menge der zulässigen Eingaben* bezeichnet, die zum angestrebten Resultat führen; das ist die Menge aller Eingabedaten x, so dass die Systemantwort gemäß Spezifikation in B liegen muss (Bild 1.5):

$$A = \{\, x \mid x \in \mathrm{dom}(R);\ (x, y) \in R \text{ impliziert } y \in B \}$$

Bei dieser Definition ist klar: Ein korrektes System *S* liefert für jede zulässige Eingabe das angestrebte Resultat.

Von einem Bedienfehler kann man natürlich nur dann reden, wenn es überhaupt zulässige Eingaben gibt, also wenn *A* nicht leer ist. Ein Bedienfehler ist dann eine Bedienhandlung, die zu einem nicht spezifikationsgemäßen Wert x' der Eingangsgröße führt ($x' \notin A$). Bei dieser Bedienhandlung kann nicht garantiert werden, dass das angestrebte Ziel erreicht wird. Was das heißt, zeigt das Beispiel

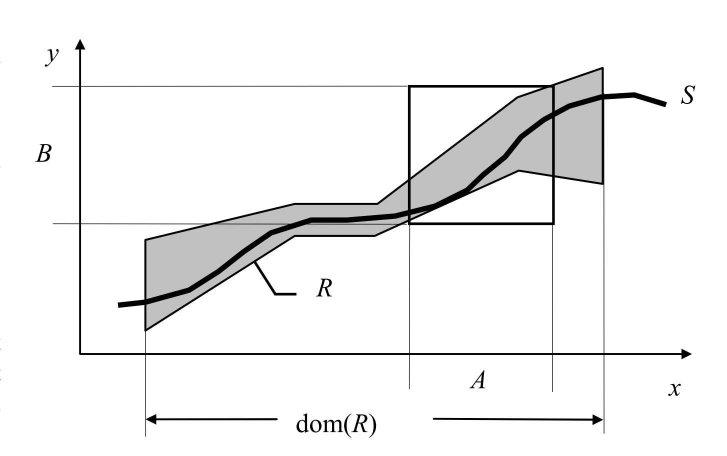


Bild 1.5 Sicherheitsspezifikation, erwünschtes Resultat *B* und zulässige Eingaben *A*

Waschmaschine. Feinwäsche soll bei 30° gewaschen werden. Wird der Schalter entsprechend eingestellt und die Wäsche dennoch gekocht, liegt wohl ein technischer Fehler vor. Ist dagegen die Waschmaschine in Ordnung und der Programmwahlschalter wird irrtümlich auf den 95° der vorangegangenen Kochwäsche belassen, ist ein Bedienfehler Ursache des unerwünschten Ergebnisses.

1.5 Die Sonderrolle der Spezifikation

Den Bedienfehlern können beobachtbare Sachverhalte zugeordnet werden, nämlich die Abweichungen von einem *normativen Modell*. Es handelt sich also um eine *operationale* Definition. Auch technische Fehler sind in diesem Sinne operational. Die normativen Modelle basieren auf der Spezifikation.

Die Operationalisierbarkeit der Begriffe stößt auf Grenzen. Der Spezifikationsfehler beispielsweise lässt sich nicht operationalisieren, indem auf ein übergeordnetes und umfassendes normatives Modell zugegriffen wird. Die Spezifikation soll ja so formuliert sein, dass *fehlerfreie* Produkte wirtschaftlich produziert werden können. Sie soll ferner gewährleisten, dass sich die Produkte leicht bedienen lassen, also keine *Bedienfehler* provozieren.

All das lässt sich erst erkennen, wenn das Produkt tatsächlich gebaut worden ist und wenn es der Bewährungsprobe auf dem Markt ausgesetzt ist. Folglich müssten umfassende normative Vorgaben für die Spezifikation bereits auf die Auswirkungen eben dieser Spezifikation vorgreifen.

Popper (1982, S. 396) merkt dazu an, dass sich Begriffsgebäude nicht konsequent aus lauter operationalen Definitionen aufbauen lassen, weil „solche Definitionen ihrem Wesen nach zirkelhaft sind".

Dieser philosophische Einwand wird durch praktische Erwägungen gestützt: Die Frage, ob eine zuverlässige und sichere Maschine wirtschaftlich realisierbar und auch bedienbar ist, lässt sich nicht vollständig im Rahmen umfassender Vorgaben beantworten. Meist wird die Antwort erst im Nachhinein geliefert, durch die Erfahrungen mit Maschine und Spezifikation. Es gibt bei-

spielsweise keine geschlossenen Modelle für die Wirkungen eines Designs auf den Anwender. Vieles, was dem Anwender anfangs ungewohnt ist, wird ihm bald vertraut; er beherrscht es schließlich. Letztlich muss der Markt die Entscheidung treffen.

Zur Vermeidung einer Zirkeldefinition wird der Spezifikation eine *Sonderrolle* zugewiesen. Die Spezifikation wird als eine *Festsetzung* betrachtet, die nicht aus übergeordneten normativen Vorgaben hergeleitet werden kann. Grundsätzlich ist die Spezifikation Ergebnis eines evolutionären Prozesses. Es sind gerade die nicht genormten Freiräume, die kreatives Schaffen ermöglichen!

Was wir - sozusagen ersatzweise - von der Spezifikation verlangen, ist *Bewährung* (Popper, 1982): Die Analyse und die Praxiserprobung der Maschine oder deren Modell müssen zeigen, dass zuverlässige und sichere Maschinen nach der Spezifikation wirtschaftlich realisierbar sind, und dass Bedienfehler ausreichend selten und unschädlich sind.

Wenn es auch keine *umfassenden* normativen Vorgaben für die Spezifikation geben kann: Es gibt einen Gestaltungsrahmen, den die geltenden Gesetze, Richtlinien und technischen Regeln sowie die bisherigen Erfahrungen bilden.

1.6 Einfache Systeme und deren Spezifikationen

Es werden nun einige Beispiele für Systeme auf unterschiedlichen Abstraktionsebenen vorgestellt. Sie sind einfach gehalten und aus unterschiedlichen Bereichen gewählt - letzteres, um die Anwendungsbreite der Zuverlässigkeitsmodellierung deutlich zu machen. Die Beispielsysteme werden hier zunächst anhand einer Spezifikation eingeführt. Später werden die Begriffe und Methoden des allgemeinen Zuverlässigkeitsmodells an diesen Beispielen erklärt.

Der Integrierer ist ein Rechenverstärker, dessen Eingangsspannung $u_E(t)$ und Ausgangsspannung $u_A(t)$ im Idealfall für $0 \leq t$ folgende Beziehung erfüllen (Hütte, 1991, G 121):

$$u_A(t) = u_A(0) - \int_0^t u_E(s)\,\mathrm{d}s$$

Bei Realisierungen solcher Rechenverstärker sind Abweichungen vom Idealfall unvermeidbar. Es seien Abweichungen des Ausgangssignals zugelassen, die unter bestimmten Umgebungsbedingungen durch die Funktion $a + bt$ beschränkt sind. Hierin sind a und b reelle Zahlen größer null. Diese Bedingungen brauchen auch nicht für alle Zeiten t eingehalten zu werden, sondern nur für ein Intervall der Dauer T. Für zulässige Eingangsgrößen soll das System also eine Ausgangsgröße $u_A(t)$ liefern, das im *Beobachtungsintervall* $[0, T]$ die Bedingung

$$| u_A(t) - u_A(0) + \int_0^t u_E(s)\,\mathrm{d}s | \leq a + bt \qquad (*)$$

erfüllt.

Nun kann man in der Praxis nicht verlangen, dass der Integrierer für jeden Verlauf der Eingangsgröße $u_E(t)$ die obigen Bedingungen erfüllt: Man wird den Wertebereich des Eingangssignals hinsichtlich der Spitzenwerte und der Frequenz so einschränken, dass die Bauelemente wie erwartet arbeiten. Man wird sie also beispielsweise in ihrem Linearitätsbereich und innerhalb ihrer Bandbreite betreiben. Das sind dann die zulässigen Beanspruchungen bzw. Eingabewerte.

Wir bezeichnen hier einmal den gesamten Verlauf einer Funktion z im vorgegebenen Beobachtungsintervall von 0 bis T mit $[z(t)]$, wohingegen $z(t)$ den Wert der Funktion zu einer gege-

benen Zeit t des Beobachtungsintervalls meint. Für die Eingangsgrößen des Systems lässt sich dann auch schreiben $x = ([u_E(t)], u_A(0))$, da sich sowohl der Kurvenverlauf der Eingangsgröße als auch der Anfangswert der Ausgangsgröße - innerhalb der gemachten Beschränkungen - frei wählen lassen. Die Ausgangsgröße des Systems ist $y = [u_A(t)]$.

Die Spezifikation R für den Integrierer ist eine Menge von Paaren (x, y) derart, dass die Beschränkungen der Eingangsgröße eingehalten werden, und dass für jede gegebene Eingangsgröße x die Ausgangsgröße y die Bedingung (*) erfüllt.

Die Funktionsprozedur Orthogonal besitzt folgende Spezifikation: Gegeben sind zwei Geraden. Die erste bildet mit der x-Achse den Winkel α und die zweite bildet mit der x-Achse den Winkel β. Die Winkel sind ganzzahlige Werte in Grad. Die Funktionsprozedur

```
FUNCTION Orthogonal(Alpha, Beta: INTEGER): BOOLEAN
```

soll den Wert TRUE liefern, wenn die beiden Geraden senkrecht aufeinander stehen, und ansonsten den Wert FALSE.

Fehlertoleranz durch Parallelredundanz wird durch ein System mit N Subsystemen (Komponenten) realisiert, die eine gemeinsame Spezifikation besitzen. Die Subsysteme werden parallel mit denselben Eingabedaten versorgt. Ein Vergleicher wertet die Ausgabegrößen der Subsysteme aus und gibt die Antwort des Gesamtsystems aus, Bild 1.6.

Spezifikation des Vergleichers: Falls die Ausgangsgrößen $y_1, y_2, ..., y_N$ der Subsysteme mehrheitlich denselben Wert haben, liefert der Vergleicher genau diesen Wert y als Ausgangsgröße des Gesamtsystems. Findet sich keine Mehrheit, ist das Ausgangssignal undefiniert. Der Fehlerausgang f liefert genau dann eine Fehlermeldung, wenn wenigstens zwei Ausgangsgrößen der Subsysteme nicht übereinstimmen.

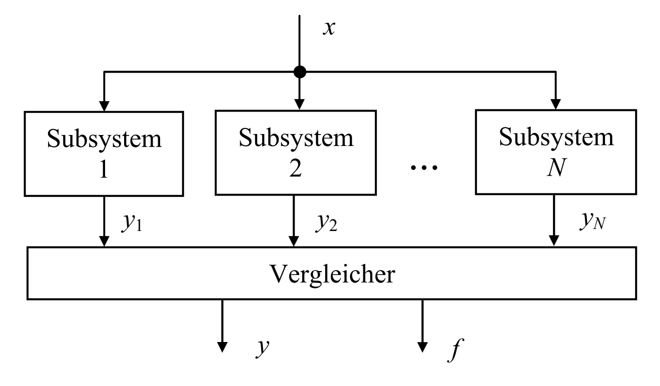


Bild 1.6 Parallelredundanz

Ein solches System heißt M-aus-N-System, wenn die Funktionsfähigkeit von wenigstens M Subsystemen vorausgesetzt wird.

Jedes System mit $M > N/2$ toleriert Fehler in bis zu N-M Subsystemen: Liefern höchstens N-M Subsysteme falsche Ausgangswerte, so werden diese Fehler mit Hilfe der anderen Subsysteme garantiert erkannt und korrigiert. Weiterhin werden alle Fehler gemeldet, solange nicht alle Subsysteme in gleicher Weise fehlerhaft antworten.

Um das immerhin noch erkannte und damit kontrollierbare Nichtfunktionieren vom gefährlichen - weil nicht gemeldeten - Nichtfunktionieren zu unterscheiden, ist hier neben einer *funktionalen Spezifikation* eine *Sicherheitsspezifikation* gegeben.

Funktionale Spezifikation: Das Gesamtsystem erfüllt dieselbe Spezifikation wie die Subsysteme und darüber hinaus wird eine Fehlermeldung ausgegeben, wenn wenigstens eins der Subsysteme nicht funktioniert.

Sicherheitsspezifikation: Das System liefert entweder Ergebnisse (Ausgabewerte) entsprechend der funktionalen Spezifikation, oder es gibt eine Fehlermeldung ab. Diese Fehlermeldung kann für die Abschaltung oder das Überführen der Anlage in einen gefahrlosen Zustand genutzt werden (Fail-Safe-Verhalten).

Beispiele: Das 2-aus-3-System toleriert Einfachfehler und meldet das Vorliegen von Ein- und Zweifachfehlern. Die funktionale Spezifikation ist erfüllt, wenn wenigstens zwei der Subsysteme intakt sind. Einfache Fehler werden also toleriert. Bezüglich der Sicherheitsspezifikation werden gar bis zu zwei fehlerhafte Subsysteme toleriert.

Das 1-aus-2-System ist speziell für die Erfüllung der Sicherheitsspezifikation gedacht. Es erfüllt die Sicherheitsspezifikation auch dann noch, wenn eines der Subsysteme defekt ist. Bezüglich der funktionalen Spezifikation liegt keinerlei Fehlertoleranz vor.

M-aus-*N*-Systeme dienen der Tolerierung bzw. Erkennung von Ausfällen in Hardware-Systemen. Gegen Fehler in verschiedenen Subsystemen mit gemeinsamer Fehlerursache, sogenannte *Common-Cause-Fehler*, hilft die Methode nicht. Unabhängigkeit der Ausfälle in den Subsystemen wird vorausgesetzt. Vorausgesetzt wird ferner, dass die Hardware ursprünglich korrekt ist. Die Subsysteme können durchaus verschiedene Exemplare eines Musters sein, also beispielsweise aus einer Fertigungsserie stammen.

Eine weitere Anwendung der *M*-aus-*N*-Systeme ist das sogenannte *Multi-Versionen-Programmieren*: Die Subsysteme sind in diesem Fall Programme, die von verschiedenen Programmierteams nach derselben Spezifikation erstellt worden sind. Hier geht es um die Erkennung von Entwurfsfehlern. Dabei unterscheidet man zwei grundlegende Strategien (Saglietti, Ehrenberger, Kersken, 1992).

- *Zufällige Diversität:* Die Programmierteams arbeiten mit denselben Vorgaben unabhängig und getrennt voneinander, um gemeinsame Irrwege möglichst auszuschließen.
- *Erzwungene Diversität:* Durch unterschiedliche Realisierungsvorgaben (Algorithmen, Programmiersprache usw.) wird die Wahrscheinlichkeit gemeinsamer Fehler zusätzlich verringert.

1.7 Qualitäts- und Zuverlässigkeitssicherung als Managementziel

Der Begriff *Qualität* wird heute sehr umfassend definiert. Er beinhaltet die Gesamtheit der Merkmale und Merkmalswerte eines Produktes oder einer Dienstleistung bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen. Zu den Qualitätsmerkmalen eines Produkts zählen

1. Funktionsfähigkeit (Korrektheit),
2. Brauchbarkeit (Bedienbarkeit als Gegensatz zur Bedienfehleranfälligkeit),
3. Zuverlässigkeit (Korrektheit auf Dauer),
4. Verfügbarkeit (Korrektheit zur Zeit),
5. Instandhaltbarkeit, Wartbarkeit,
6. Sicherheit,
7. Umweltverträglichkeit,
8. Wirtschaftlichkeit (Effizienz),
9. Schönheit.

Je nach Produkt kommen dann noch spezifische Merkmale hinzu. Beispielsweise zeichnet sich gute Software dadurch aus, dass sie auf verschiedenen Plattformen (Hard- und Software-Umgebungen) funktioniert, und dass sie sich leicht an weitere Aufgaben anpassen lässt. Ihre wesentlichen Merkmale sind

10. Übertragbarkeit (Portabilität),

11. Wiederverwendbarkeit,
12. Erweiterbarkeit,
13. Änderbarkeit.

Dazu kommen die mittelbaren Qualitätsmerkmale, die im Dienste der oben formulierten stehen:

14. Einfache Struktur, Verständlichkeit, Lesbarkeit des Programmtextes,
15. Prüfbarkeit (Einfachheit des Zuverlässigkeits-/Korrektheitsnachweises).

Ein wichtiger Maßstab der Qualität ist die Kundenzufriedenheit.

Die grundlegenden Normen zum Thema Qualität und Qualitätsmanagement sind DIN 55 350, DIN EN ISO 8402 und DIN EN ISO 9000 bis 9004.

Bisher haben wir Merkmale eines Produktes vor allem auf den Gebrauch oder die Anwendung bezogen. Aber der Gebrauch ist nur eine Phase des gesamten Produktlebenszyklus (Hütte, 1991; Zemanek, 1992); siehe den Kasten "Produktlebensphasen".

Mit TQM wird heute das *umfassende Qualitätsmanagement* (Total Quality Management) bezeichnet: TQM ist die Managementmethode einer Organisation, die

- funktionsübergreifend angelegt ist (Logistik, Entwicklung, Konstruktion, Fertigung, Marketing, Controlling),
- alle Phasen des Produktlebenszyklus berücksichtigt,
- Qualität in den Mittelpunkt stellt,
- über die Kundenzufriedenheit den langfristigen Geschäftserfolg und Nutzen für die Organisation und für die Gesellschaft anstrebt.

Die Erreichung dieser Ziele wird gefördert durch

- überzeugende und nachhaltige Führung der obersten Leitung,
- Ausbildung und Schulung der Mitglieder in allen Stellen und Hierarchieebenen der Organisation.

TQM basiert auf der Mitwirkung aller Mitglieder der Organisation. Für einen Produktionsbetrieb leitet sich daraus unter anderem ab, dass grundsätzlich jeder Ersteller einer Leistung die Qualitätsverantwortung trägt. Der Maschinen- und Prozessbediener sichert die von ihm verantwortete Qualität in einem kleinen Prozess-Regelkreis und führt die nötigen Korrekturen möglichst direkt durch. Dem überlagern sich weitere Qualitätsregelkreise bis hin zum großen Produkt-Regelkreis, der auch die Erfahrung im Feldeinsatz beim Kunden erfasst (Hering/Triemel/Blank, 1996).

Produktlebensphasen

1. Anforderungserfassung
   Ergebnis: verbale Beschreibung des Produkts, nichtformale Spezifikation
2. Spezifizierung
   Ergebnis: *formale Spezifikation*
3. Entwurf
   Ergebnis: Architektur des Produkts (z.B.: Blockschaltbild)
4. Implementierung
   Ergebnis: Konstruktionsunterlagen
5. Realisierung
   Ergebnis: Fertigungsunterlagen (z.B.: Stücklisten)
6. Fertigung/Montage/Prüfung
7. Vertrieb/Beratung/Verkauf
8. Gebrauch/Verbrauch/Wartung
9. Recycling/Entsorgung

2 Wahrscheinlichkeitsrechnung und Statistik

Die folgende kurze Zusammenstellung des benötigten Stoffes aus Wahrscheinlichkeitsrechnung und mathematischer Statistik dient in erster Linie der Klarstellung des Gebrauchs der Symbole. Ausführliche Darstellungen enthalten beispielsweise die Bücher von Fisz (1976) und Sachs (1992).

2.1 Elementare Wahrscheinlichkeitsrechnung

Wir betrachten Zufallsergebnisse, wie sie unter anderem beim Wurf eines Würfels auftreten. Das Zufallsergebnis bezeichnen wir einmal mit ζ. Die Menge der Werte, die die Variable ζ annehmen kann, ist der Ereignisraum Ω. Im Falle des Würfels besteht der Ereignisraum aus den Zahlen 1 bis 6, also $\Omega = \{1, 2, 3, 4, 5, 6\}$. Die Teilmengen des Ereignisraums heißen Ereignisse, die Elemente des Ereignisraums sind die Elementarereignisse.

Seien *A, B, C, ...* Ereignisse. Diese werden durch logische Bedingungen oder die explizite Angabe der Mengen dargestellt: $\zeta < 5$ steht im Falle des Würfels beispielsweise für die Menge $\{1, 2, 3, 4\}$. Eine Bedingung wird also mit ihrer *Erfüllungsmenge* – das ist die Menge der Werte, die diese Bedingung erfüllen - gleich gesetzt. Für die Vereinigung zweier Ereignisse A und B wird $A \cup B$ geschrieben und ihr Durchschnitt ist AB.

Die Ereignisse treten mit gewissen Wahrscheinlichkeiten ein. Die Elementarereignisse eines Wurfs mit einem gerechten Würfel haben alle die Wahrscheinlichkeit 1/6. Die tatsächlichen Wahrscheinlichkeiten für einen realen Würfel wird man über eine große Anzahl von Versuchen näherungsweise aus den relativen Häufigkeiten der einzelnen Punktzahlen ermitteln. Für die Wahrscheinlichkeit eines Ereignisses A schreiben wir $P(A)$.

Aus der Häufigkeitsdefinition der Wahrscheinlichkeiten lassen sich direkt die fundamentalen Regeln für das Rechnen mit Wahrscheinlichkeiten herleiten:

1. Für jedes Ereignis A gilt $0 \le P(A) \le 1$.
2. Die Wahrscheinlichkeit des unmöglichen Ereignisses ist gleich null: $P(\{\}) = 0$.
3. Die Wahrscheinlichkeit des sicheren Ereignisses ist gleich eins: $P(\Omega) = 1$.
4. Für zwei einander ausschließende Ereignisse A und B, wenn also $AB = \{\}$ ist, gilt die Regel der Additivität: $P(A \cup B) = P(A) + P(B)$.

Aus diesen Regeln lassen sich weitere gewinnen. Beispielsweise die folgende für zwei sich nicht notwendigerweise ausschließende Ereignisse A und B: $P(A \cup B) = P(A)+P(B)-P(AB)$.

Mit $-A$ bezeichnen wir das zu A komplementäre Ereignis: $A\text{-}A = \{\}$, $A \cup -A = \Omega$.

Die bedingte Wahrscheinlichkeit des Ereignisses A unter der Bedingung B ist definiert durch

$$P(A|B) = \frac{p(AB)}{p(B)}.$$

Daraus folgt unter anderem die Formel

$$\frac{p(A \mid B)}{p(A)} = \frac{p(B \mid A)}{p(B)}.$$

Wenn man darin noch $p(B)$ durch $p(B|A)\cdot p(A)+p(B|-A)\cdot p(-A)$ ersetzt, erhält man die *Formel von Bayes*.

Sei $p(A|B)$ die Wahrscheinlichkeit des Sachverhalts A unter der Bedingung der Beobachtung B und dementsprechend $p(B|A)$ die Wahrscheinlichkeit der Beobachtung B bei gegebenem Sachverhalt A. Die obige Formel besagt dann: Ein Sachverhalt A wird durch die Beobachtung B um denselben Faktor wahrscheinlicher wie die Beobachtung B durch den Sachverhalt A wahrscheinlicher wird. In diesem Zusammenhang nennt man $p(A)$ die A-priori-Wahrscheinlichkeit des Sachverhalts; $p(A|B)$ ist seine A-posteriori-Wahrscheinlichkeit.

2.2 Zufallsvariable

Zufallsvariable sind reellwertige Zufallsergebnisse. Sie sind durch ihre *Verteilungsfunktion* charakterisiert. Die Verteilungsfunktion F einer Zufallsvariablen X ist definiert durch

$$F(x) = P(X < x).$$

Dabei ist $P(X < x)$ die Wahrscheinlichkeit dafür, dass der Wert der Variablen kleiner als x ist. Die Verteilungsfunktion ist monoton wachsend und es gilt $F(-\infty) = 0$ und $F(\infty) = 1$. Die Wahrscheinlichkeit, dass der Wert der Variablen in das Intervall $[a, b)$ fällt, ist gegeben durch

$$F(b) - F(a).$$

Sei X eine *diskrete Zufallsvariable*, die die Werte x_i ($i = 1, 2, 3, ..., n$) annehmen kann, und p_i die *Wahrscheinlichkeit* dafür, dass $X = x_i$ ist. Die Verteilungsfunktion ist dann gegeben durch

$$F(x) = \sum\nolimits_{i|x_i<x} p_i \ .$$

Stetige Zufallsvariable zeichnen sich dadurch aus, dass es eine *Dichte* $f(x)$ gibt, so dass die Verteilungsfunktion $F(x)$ sich als deren Integral darstellen lässt:

$$F(x) = \int_{-\infty}^{x} f(u)\,\mathrm{d}u \ .$$

Die wichtigsten Formeln und Kennzahlen seien hier für die diskreten Zufallsvariablen kurz in Erinnerung gerufen. Der *Erwartungswert* (*Mittelwert*) einer diskreten Zufallsvariablen X ist gegeben durch

$$\mu = E[X] = \sum_{i=1}^{n} x_i p_i \ .$$

Sei g eine reelle Funktion und X eine Zufallsvariable, dann definiert $g(X)$ ebenfalls eine Zufallsvariable. Ihr Erwartungswert ist gegeben durch

$$E[g(X)] = \sum_{i=1}^{n} g(x_i) p_i \ .$$

Ist die Zufallsvariable stetig, ist ihr Erwartungswert gleich

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x)\,\mathrm{d}x \ .$$

Mit diesen Formeln lässt sich die *Varianz* (auch: *Streuung*) einer Zufallsvariablen ausrechnen. Die Varianz σ^2 einer Zufallsvariablen X ist definiert durch

$$\sigma^2 = E[(X - \mu)^2] = E[X^2] - \mu^2.$$

Der Wert σ ist die *Standardabweichung* der Zufallsvariablen.

Hat die Zufallsvariable X den Erwartungswert μ und die Standardabweichung σ, dann hat die durch die *lineare Transformation* $Y = aX + b$ definierte Zufallsvariable Y den Erwartungswert $a\mu + b$ und die Standardabweichung $|a|\sigma$.

Seien nun X und Y zwei Zufallsvariable, die die Werte x_i (i = 1, 2, ..., n) bzw. y_j (j = 1, 2, ..., m) mit den Wahrscheinlichkeiten p_i bzw. q_j annehmen. Die Wahrscheinlichkeit dafür, dass $X = x_i$ und $Y = y_j$ bezeichnen wir mit p_{ij}.

Offensichtlich bestehen zwischen den Wahrscheinlichkeiten die Beziehungen $p_i = \sum_j p_{ij}$ und $q_j = \sum_i p_{ij}$. Sind die Zufallsvariablen *statistisch unabhängig*, dann gilt: $p_{ij} = p_i\, q_j$.

Der Erwartungswert der Summe von Zufallsvariablen ist gleich der Summe der Erwartungswerte: $E[X + Y] = \Sigma_{ij} (x_i + y_j)\, p_{ij} = \Sigma_{ij}\, x_i\, p_{ij} + \Sigma_{ij}\, y_j\, p_{ij} = \Sigma_i\, x_i\, p_i + \Sigma_j\, y_j\, q_j = E[X] + E[Y]$.
Bei unabhängigen Zufallsvariablen gilt eine entsprechende Formel auch für das Produkt: $E[X{\cdot}Y] = E[X]{\cdot}E[Y]$. In diesem Fall ist die Varianz additiv: $\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2$.

2.3 Verteilungen

Normalverteilung: Die Dichte der *Normalverteilung* ist gegeben durch

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} .$$

Die Parameter μ und σ in dieser Formel haben zugleich die Bedeutung von Erwartungswert und Standardabweichung. Eine Zufallsvariable mit dieser Verteilung bezeichnet man kurz als (μ, σ)-normalverteilt. Es handelt sich bei $f(x)$ um die berühmte Glockenkurve, deren Maximum bei μ liegt. Die Wendepunkte befinden sich jeweils im Abstand σ von der Maximumstelle.

Die herausragende Bedeutung der Normalverteilung rührt daher, dass viele Zufallsgrößen, die in der Natur beobachtet werden können, tatsächlich näherungsweise normalverteilt sind. Das lässt sich darauf zurückführen, dass diese Zufallsvariablen aus der Überlagerung vieler einzelner, weitgehend voneinander unabhängiger Einflüsse entstehen. Und eine Summe von vielen unabhängigen Zufallsvariablen gleicher Größenordnung ist tatsächlich annähernd normalverteilt. Die Näherung ist umso genauer, je größer die Anzahl der Summanden ist. Diesen Sachverhalt bezeichnet man als den *zentralen Grenzwertsatz* der mathematischen Statistik. Am Beispiel der Binomialverteilung wird er in Bild 2.2 demonstriert. Mathematische Präzisierungen liegen in verschiedenen speziellen Grenzwertsätzen vor.

Exponentialverteilung: Zufallsvariable ist eine Zeit T bis zum Eintritt eines Ereignisses, das unvorhersehbar ist, wie beispielsweise der Zerfall eines Atoms. Eine solche zufällige Zeit heißt *exponentialverteilt* wenn sie die folgende Eigenschaft besitzt:

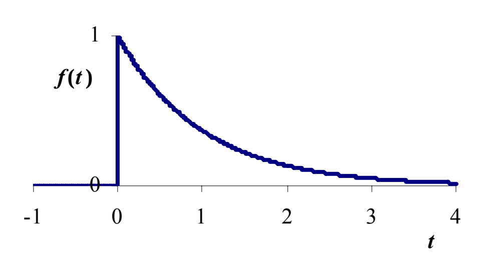


Bild 2.1 Dichte der Exponentialverteilung für λ=1

Die Wahrscheinlichkeit für das Auftreten des Ereignisses in einem Intervall $[t, t+\mathrm{d}t)$ - unter der Bedingung, dass es bis dahin noch nicht aufgetreten ist - ist unabhängig von der bis dahin verstrichenen Zeit t. Außerdem ist diese Wahrscheinlichkeit für hinreichend kleine Intervalle proportional zur Intervalllänge dt.

Bezeichnet man die angesprochene Proportionalitätskonstante mit λ, so ist die Verteilungsfunktion der Exponentialverteilung gegeben durch

$$F(t) = 0 \text{ für } t < 0, \text{ und}$$

$$F(t) = 1 - e^{-\lambda t} \text{ für } 0 \le t$$

Die zugehörige Verteilungsdichte ist gleich

$$f(t) = \lambda\, e^{-\lambda t} \text{ für } 0 \le t.$$

Erwartungswert und Standardabweichung dieser Verteilung sind jeweils gleich $1/\lambda$.

Dass die Exponentialverteilung tatsächlich die eben angesprochene Eigenschaft hat, soll kurz nachgewiesen werden: Die Wahrscheinlichkeit, dass das Ereignis in das - als hinreichend klein angenommene - Intervall $[t, t+\mathrm{d}t)$ fällt, ist annähernd gleich $f(t)\cdot\mathrm{d}t$, also gleich $\lambda\cdot e^{-\lambda t}\cdot\mathrm{d}t$. Die Wahrscheinlichkeit dafür, dass das Ereignis bis zum Zeitpunkt t noch nicht eingetreten ist, ist gleich $1\text{-}F(t)$, also gleich $e^{-\lambda t}$. Die bedingte Wahrscheinlichkeit dafür, dass das Ereignis im genannten Intervall eintritt, unter der Bedingung, dass es vorher noch nicht eingetreten ist, ist gleich dem Quotienten $f(t)\cdot\mathrm{d}t/(1\text{-}F(t))$ dieser Werte, also gleich $\lambda\cdot\mathrm{d}t$. Das war zu zeigen: Die fragliche Wahrscheinlichkeit ist unabhängig von t und proportional zu dt.

Binomialverteilung: Bei der *Binomialverteilung* gehen wir von einem Versuchsschema aus, das aus einer Folge von n Versuchen besteht. Jeder der Versuche hat eines von zwei möglichen Ergebnissen, und zwar tritt das Ereignis A ein oder das dazu komplementäre Ereignis B. Die Wahrscheinlichkeit für das Ereignis A sei p. Die Wahrscheinlichkeit des Ereignisses B ist dann gleich $1 - p$. Die Ergebnisse der Versuche seien unabhängig voneinander.

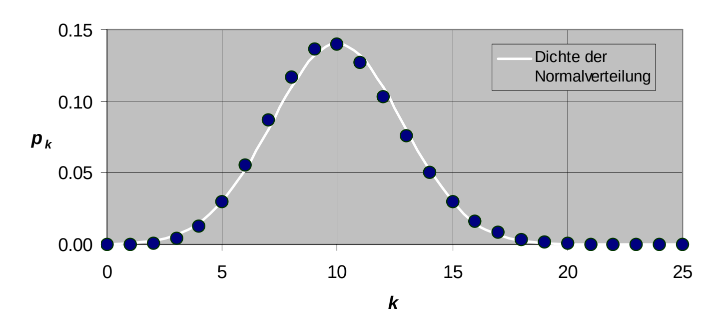


Bild 2.2 Wahrscheinlichkeiten der (50, 0.2)-Binomialverteilung

Mit X_i bezeichnen wir die Zufallszahl, die den Wert 1 annimmt, wenn im i-ten Versuch Ereignis A eingetreten ist; im Fall des Ereignisses B hat sie den Wert 0. Mit X bezeichnen wir die Summe der Zufallsvariablen vom ersten bis zum n-ten Versuch: $X = X_1 + X_2 + ... + X_n$. Die Wahrscheinlichkeit dafür, dass das Ereignis A in der Versuchsserie genau k mal eintritt (und das Ereignis B demzufolge n-k mal), ist gleich

$$p_k = P(X = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k}.$$

Die Verteilungsfunktion der (n, p)-Binomialverteilung ist

$$F(x) = P(X < x) = \sum_{k<x} \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} \ .$$

Der Erwartungswert der (n, p)-Binomialverteilung ist gegeben durch $\mu = n\cdot p$ und die Standardabweichung durch $\sigma = \sqrt{n \cdot p \cdot (1-p)}$.

Bild 2.2 zeigt die Wahrscheinlichkeiten der (50, 0.2)-Binomialverteilung. Die Binomialverteilung ist die Summe unabhängiger Zufallsvariablen. Diese Summe wird wegen des zentralen Grenzwertsatzes durch die Normalverteilung mit demselben Erwartungswert von 10 und derselben Standardabweichung von 2.83 gut angenähert.

2.4 Vertrauensintervalle

Wir haben für die Ergebnisvariable X die N Stichprobenwerte $x_1, x_2, ..., x_N$ erhalten. Wir interessieren uns für den Erwartungswert $\mu = E[X]$ der Ergebnisvariablen und nehmen das arithmetische Mittel m der Stichprobenwerte als Schätzwert für μ:

$$m = (\sum_{i=1}^{N} x_i) / N$$

Es fragt sich nun, inwieweit man damit rechnen kann, dass $m \approx \mu$ ist, und welchen Einfluss der Stichprobenumfang auf die Güte des Schätzwerts hat. Präziser ausgedrückt geht es darum, den Stichprobenumfang N zu bestimmen, so dass mit einer vorgegebenen *Sicherheit Q* (in %) der absolute Fehler nicht größer als die *Fehlergrenze E* ist:

$$|m - \mu| \leq E$$

Der Schätzwert m ist selbst eine Zufallsvariable: Wir fassen die x_i als Realisierungen von Zufallsvariablen X_i auf, die alle dieselbe Verteilung wie X besitzen, und die voneinander statistisch unabhängig sind.

Der Erwartungswert der Zufallsvariablen m ist gleich μ. Der Schätzwert hat demnach wenigstens den richtigen Erwartungswert. Die Standardabweichung ist gleich $\sigma / \sqrt{N}$. Sie nimmt mit wachsendem N ab. Das heißt, dass die Schätzwerte umso weniger streuen, je größer die Stichprobe ist. Die Sicherheit, dass m in der Nähe von μ liegt, nimmt mit dem Stichprobenumfang zu.

Diese Aussage lässt sich folgendermaßen quantifizieren. In der Praxis ist N meist genügend groß, so dass aufgrund des zentralen Grenzwertsatzes die Zufallsvariable $\sqrt{N}(m - \mu)/\sigma$ als näherungsweise (0, 1)-normalverteilt angesehen werden kann. Die Dichtefunktion der Normalverteilung mit dem Erwartungswert 0 und der Standardabweichung 1 ist gegeben durch

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2}} .$$

Der Wert t wird zu einem vorgegebenen Q so bestimmt, dass

$$Q = \int_{-t}^{t} \Phi(x) \mathrm{d}x$$

gilt. Die wichtigsten Wertepaare t und Q sind in der Tabelle 2.1 aufgelistet. Die genannten Zusammenhänge führen zur Formel

$$P(|m-\mu| \leq t\,\sigma / \sqrt{N}\,) = Q.$$

Tabelle 2.1 Gebräuchliche Werte für Q und t

Q	t
60 %	0.84
68.27 %	**1**
80 %	1.28
90 %	1.64
95 %	1.96
95.5 %	**2**
98 %	2.33
99 %	2.58
99.7 %	**3**

Der absolute Fehler $|m-\mu|$ ist mit der Wahrscheinlichkeit Q durch $E = t\sigma/\sqrt{N}$ begrenzt. Das heißt: Der korrekte Wert μ liegt mit der *Sicherheit* (auch: Vertrauenswahrscheinlichkeit) Q im Intervall $[m - E, m + E]$. Dieses zufällige Intervall heißt *Konfidenzintervall* oder *Vertrauensintervall*. Für das Konfidenzintervall schreiben wir manchmal auch kurz $m \pm E$.

Da $\sigma/\sqrt{N}$ gleich der Standardabweichung der Schätzgröße m ist, bezeichnet man das Konfidenzintervall $m \pm t\sigma/\sqrt{N}$ auch als $t\sigma$-Bereich. Zur statistischen Sicherheit von 95.5 Prozent gehört demnach der 2σ-Bereich.

Die Methode der Bestimmung des Vertrauensintervalls hat noch einen Schönheitsfehler: Wir kennen σ im Allgemeinen nicht. Wir verschaffen uns für σ einen Schätzwert s, indem wir dieselbe Stichprobe wie zur Bestimmung von μ benutzen. Wir setzen

$$s^2 = \frac{1}{N-1} \cdot \sum_{i=1}^{N} (x_i - m)^2 = \frac{1}{N-1} \cdot (\sum_{i=1}^{N} x_i^2 - Nm^2)$$

und stellen fest, dass s^2 den Erwartungswert σ^2 besitzt.

Eigentlich ist s auch nur ein Schätzwert für σ. Erneut stellt sich die Frage nach dem Vertrauensintervall. Ein Teufelskreis kündigt sich an. Wie man ihm entkommen kann, findet man in den Lehrbüchern der Statistik. Hier nur so viel: Bei Stichprobengrößen ab etwa 40 darf man s anstelle von σ in die Formel für die Fehlergrenze einsetzen: $E = ts/\sqrt{N}$ für $N \geq 40$.

Für die Genauigkeitsbetrachtung dieses Abschnitts wurde vorausgesetzt, dass

- die Stichprobenwerte voneinander statistisch unabhängig sind,
- allen Stichprobenwerten dieselbe Verteilung zugrunde liegt,
- der Schätzwert m (näherungsweise) normalverteilt ist, und
- s ein guter Schätzwert für σ ist.

Die letzten beiden Voraussetzungen sehen wir als erfüllt an, wenn $N \geq 40$.

3 Zuverlässigkeit einfacher Systeme

3.1 Festigkeit, Last, Ausfall

Ein Rohr wird leck, wenn der Druck zu groß wird. Ein Seil reißt, wenn die Zugkraft die Festigkeit überschreitet. Ein elektronischer Schaltkreis fällt aus, wenn die angelegte Spannung zu einem so großen Strom führt, dass aufgrund der entstehenden Hitze Halbleitermaterial schmilzt. Ein hydraulisches Ventil fällt aus, wenn die Dichtung dem Druck nicht widerstehen kann. Eine Welle bricht, sobald das auftretende Drehmoment die Festigkeit überschreitet. All dies sind Fälle, in denen es zu Ausfällen kommt, weil die Last (Load) die gegebene Festigkeit (Strength) überschreitet.

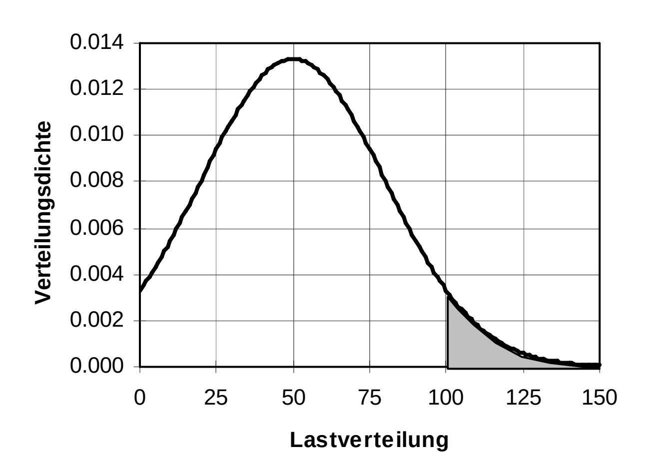


Bild 3.1 Ausfallwahrscheinlichkeit (schraffierte Fläche) bei einer (50, 30)-normalverteilten Last und einer Festigkeit von 100

Last meint also eine physikalische Größe wie Druck, Kraft, Spannung, Temperatur - während mit Festigkeit jegliche physikalische Widerstandskraft gemeint ist (z.B. mechanische Festigkeit, Schmelzpunkt). Weder Last noch Festigkeit sind im Allgemeinen konstante Größen. Sie sind statistisch verteilt. Für die Wahrscheinlichkeit, dass die Last L die Festigkeit S überschreitet - also für die Wahrscheinlichkeit des Ausfalls -, schreiben wir $P(L>S)$. Seien f_L bzw. f_S die Verteilungsdichten für Last und Festigkeit und F_L bzw. F_S die zugehörigen Verteilungsfunktionen. Die Wahrscheinlichkeit des Ausfalls ist gegeben durch

$$P(L>S) = \int_0^\infty f_S(S)\left(\int_S^\infty f_L(L)dL\right)dS = \int_0^\infty f_S(S)\left(1 - F_L(S)\right)dS\ .$$

Diese Größe ist einfach zu berechnen, wenn sowohl Last als auch Festigkeit jeweils normalverteilt sind (O´Connor, 1991, S. 104 f.) Falls die Festigkeit deterministisch und gleich S ist, geht diese Formel über in

$$P(L>S) = \int_S^\infty f_L(L)dL\ = 1 - F_L(S)\ .$$

Für eine (50, 30)-normalverteilte Last und einer Festigkeit von 100 sind die Verhältnisse in Bild 3.1 dargestellt. Das "System" erfüllt die Spezifikation, wenn die Festigkeit S wenigstens so groß wie eine geforderte Festigkeit $S_{\min}$ ist: $S_{\min} \le S$. Alle Beanspruchungen, die die gefor-

derte Festigkeit nicht übersteigen, sind zulässig: $L \leq S_{\min}$. Bei der im Bild gezeigten Lastverteilung sind unzulässige Beanspruchungen möglich, die zum Ausfall führen können.

3.2 Betriebsbelastungen und Materialermüdung

Betriebsbelastungen ergeben sich aus den zulässigen Beanspruchungen (Druck, Temperatur, Spannung usw.) und den Umweltbedingungen (Schmutz, Feuchte, Druck, Temperatur, korrodierende Atmosphäre usw.). Ein Beispiel dafür, dass die regulären Betriebsbelastungen zu Fehlern führen, bilden die Rohrleitungen, wie man sie beispielsweise in Kraftwerken und in der Energieverteilung findet (Öl-, Gas- und Fernwärmeleitungen). Hier führen Druckschwankungen zur Materialermüdung. Bauteile, die dynamischen Belastungen unterworfen sind, fallen früher aus als solche, die nur einer statischen Belastung ausgesetzt sind. Die Abhängigkeit der maximalen Zahl N von Druckschwankungen der Höhe ΔP ist mit den Konstanten C_1, C_2 und k näherungsweise darstellbar durch $N = N(\Delta P) = C_1(\Delta P/C_2)^{-k}$. Stellt man diese Funktion im doppelt-logarithmischen Maßstab grafisch dar, ergibt sich die sogenannte *Wöhlerlinie*, eine Gerade mit der Steigung $-k$. Für geschweißte Rohre gilt $k = 3.75$ und für nahtlose Rohre $k = 4.5$. Für die Temperaturbelastung von Bauteilen gelten ähnliche Beziehungen (Kuhlmann, 1981, S. 112 ff.).

3.3 Versagen und Zuverlässigkeitsfunktion

Unkorrekte Systeme erkennen wir daran, dass sie zuweilen versagen. Ein System versagt genau dann, wenn zu einem technischen Fehler eine fehleroffenbarende Beanspruchung hinzukommt, Bild 3.2. Ausgangspunkt der Zuverlässigkeitsmodellierung ist die *Zuverlässigkeitsfunktion* $Z(t)$ des betrachteten Systems. Sie ist gleich der Wahrscheinlichkeit, dass im Zeitintervall von 0 bis t das System nicht versagt.

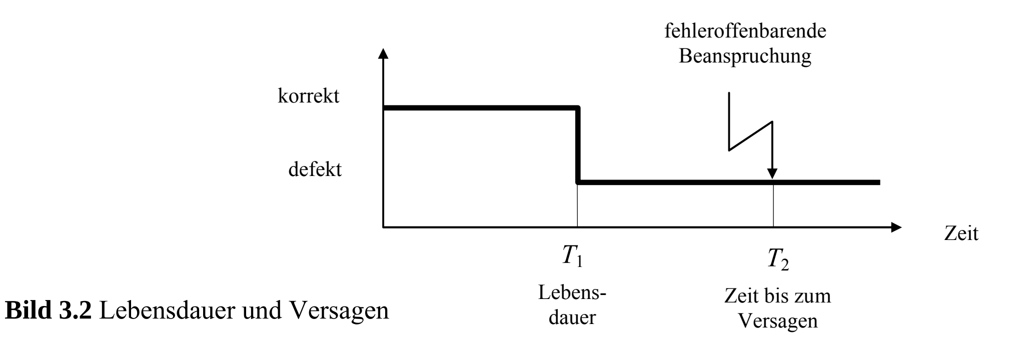


Bild 3.2 Lebensdauer und Versagen

In der Theorie der Hardware-Zuverlässigkeit geht man davon aus, dass sich jeder Ausfall sofort offenbart. Damit tritt der Ausfall an die Stelle des Versagens. Die Zuverlässigkeitsfunktion wird zur zeitabhängigen *Überlebenswahrscheinlichkeit.*

Wenn die Zeit bis zum ersten Versagen eine stetige Zufallsvariable ist, dann ist deren Verteilungsfunktion gleich $F(t) = 1\text{-}Z(t)$, und die Verteilungsdichte ist gegeben durch

$$f(t) = -\frac{\mathrm{d}Z(t)}{\mathrm{d}t} = -\dot{Z}(t)\,.$$

Der Erwartungswert der Zufallsvariablen *Zeit bis zum ersten Versagen* wird mit *MTTF* (Mean Time To Failure) bezeichnet. Es gilt

$$MTTF = \int_0^\infty Z(t)\mathrm{d}t \ .$$

Diese Formel ergibt sich, wenn man im Integral für den Erwartungswert $\int_0^\infty t \cdot f(t)\mathrm{d}t$ die Funktion $f(t)$ durch $-\dot{Z}(t)$ ersetzt und die Regel der partiellen Integration anwendet. Dabei ist noch vorauszusetzen, dass $t{\cdot}Z(t)$ mit wachsendem t gegen 0 strebt.

Die *(System-)Versagensrate* $\lambda(t)$ eignet sich zur grafischen Veranschaulichung des zu erwartenden Versagensverhaltens des Gesamtsystems. Sie hat folgende Bedeutung: Unter der Bedingung, dass das System bis zum Zeitpunkt t noch nicht versagt hat, ist die Wahrscheinlichkeit dafür, dass das System im Zeitintervall von t bis $t+\Delta t$ versagt, näherungsweise gleich $\Delta t{\cdot}\lambda(t)$. Dabei wird ein relativ kleines Δt vorausgesetzt. Die Versagensrate lässt sich aus der Zuverlässigkeitsfunktion folgendermaßen ermitteln:

$$\lambda(t) = -\frac{\mathrm{d}Z(t)/\mathrm{d}\,t}{Z(t)} = -\frac{\dot{Z}(t)}{Z(t)} = -\frac{\mathrm{d}\ln(Z(t))}{\mathrm{d}\,t} \ .$$

Die Integration dieser Gleichung liefert die Formel

$$Z(t) = e^{-\int_0^t \lambda(u)du}$$

für die Abhängigkeit der Zuverlässigkeitsfunktion von der Versagensrate. Bei konstanter Versagensrate λ geht diese Formel über in $Z(t) = e^{-\lambda t}$. Die Zeit bis zum Versagen ist dann exponentialverteilt.

In der Hardware-Zuverlässigkeit nimmt die *Ausfallrate* die Stelle der Versagensrate ein. Falls die Ausfallrate konstant ist, liegt eine exponentialverteilte *Lebensdauer* vor. Die *mittlere Lebensdauer* ist in diesem Fall gleich dem Kehrwert der Ausfallrate.

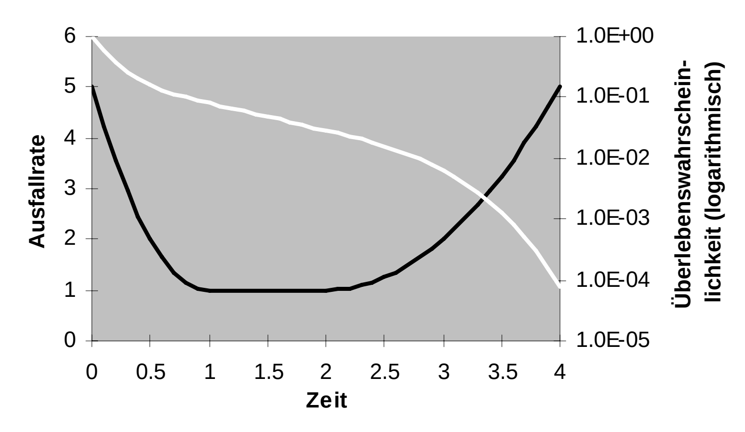


Bild 3.3 Die Badewannenkurve der Ausfallrate (dunkle Kurve) und die zugehörige Überlebenswahrscheinlichkeit (helle Kurve)

Der typische Verlauf der Ausfallrate eines Bauelements wird durch die sogenannte *Badewannenkurve* wiedergegeben (Bild 3.3): Einer Phase der Frühausfälle mit sinkender Ausfallrate schließt sich die mittlere Lebensphase mit konstanter Ausfallrate an. In der späten Lebensphase steigt die Ausfallrate aufgrund von Verschleiß und Ermüdung wieder an (DIN 4004/2).

Der Überwindung der Frühausfallphase dient das sogenannte *Burn-in:* Über eine gewisse Zeit werden - vor der eigentlichen Inbetriebnahme - die Komponenten höheren Temperaturen ausgesetzt. Dadurch sollen Ausfälle der schwachen Komponenten provoziert werden, ohne die einwandfreien zu schädigen.

3.4 Ausfallraten: Schätzungen, Daten

Für Elektronik-Bauteile ist für die Errechnung von Bauelementeausfallraten das Verfahren nach dem MIL-HDBK-217 eingeführt. Dort findet man für jeden Bauelement-Typ die passende Formel für die Ermittlung der Ausfallrate. Für Bauelemente der Mikroelektronik sieht das Modell so aus:

$$\lambda = \pi_Q \pi_L [C_1 \pi_T \pi_V + C_2 \pi_E] 10^{-6}/\text{h}.$$

Es berücksichtigt eine ganze Reihe von Einflussfaktoren (Pi-Faktoren). Die Einflussfaktoren beziehen sich auf die Umgebung (E, Environment), die Qualität (Q), die Spannungsbelastung (V, Voltage derating stress factor), den Lernfaktor (L) und die Umgebungstemperatur (T). Bei anderen Modellen gehen noch die Anwendung (A, Application) und die Konstruktion (C) in die Formeln ein; C_1 ist ein Komplexitätsfaktor, der die Anzahl der Gatter, die Anzahl der Bits bei Speichern oder die Anzahl von Transistoren bei Analogschaltungen erfasst; C_2 ist ein Komplexitätsfaktor, der Aspekte des Aufbaus der Baueinheiten wie Pinzahl und Gehäusetyp berücksichtigt (O´Connor, 1991, S. 229 ff.; Shooman, 1990, S. 637 ff.).

Ausfallraten von Kraftwerkskomponenten sind in der Tabelle 3.1 am Schluss des Kapitels zusammengestellt.

3.5 Ausfallratenaddition

Ein System möge aus *n* Komponenten mit den konstanten Ausfallraten $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n$ bestehen. Die Überlebenswahrscheinlichkeit der *k*-ten Komponente ist $Z_k(t) = e^{-\lambda_k t}$. Das System sei *nicht redundant* aufgebaut, so dass jeder Ausfall einer Komponente auch zum Ausfall des Systems führt. Andersherum: Das System überlebt genau dann, wenn sämtliche seiner Komponenten überleben. Die Ausfälle mögen *statistisch unabhängig* voneinander auftreten. Die Überlebenswahrscheinlichkeit des Gesamtsystems ergibt sich dann aus dem Produkt der Überlebenswahrscheinlichkeiten der Komponenten: $Z(t) = Z_1(t)\, Z_2(t) \ldots Z_n(t) = e^{-(\lambda_1+\lambda_2+\ldots+\lambda_n)t}$. Die Ausfallrate des Gesamtsystems ist - unter den genannten Bedingungen - also gleich der Summe der Ausfallraten der Komponenten. Dieses simple Verfahren der Ausfallratenaddition lässt sich auch noch auf gewisse redundante Systeme - insbesondere sicherheitsrelevante Schaltungen - ausdehnen (Grams/Angermann, 1981). Dazu ein Beispiel:

Elektronik. Eine Automatisierungseinrichtung möge etwa 2 000 Elektronikkarten enthalten. Die Ausfallrate einer Elektronikkarte wurde zu 10^{-6}/h veranschlagt. Die Ausfallrate der gesamten Elektronik ergibt sich zu $2 \cdot 10^{-3}$/h. Die Wahrscheinlichkeit dafür, dass innerhalb eines Tages die Anlage wegen eines Elektronik-Defekts ausfällt, ist gleich $1-e^{-0.002\cdot 24} \approx 5$ %.

3.6 Poisson-Ströme

Wir stellen uns eine Folge von Ereignissen vor. Ein solches Ereignis kann sein das Versagen eines Systems, der Ausfall eines Systems mit unmittelbar darauffolgender Wiederherstellung oder auch die Ankunft eines Kunden vor dem Schalter einer Bank, ein radioaktiver Zerfall, ein vorbeifahrendes Automobil.

Die Folge der Ereignisse, auch *Strom* genannt, bildet einen stochastischen Prozess. Wir wollen voraussetzen, dass er sich als *Markoff-Prozess* modellieren lässt (Bild 3.4). Bei einem solchen Prozess unterscheiden wir verschiedene *Zustände*, die als Kreise dargestellt werden. Die Übergänge zwischen den Zuständen werden als Pfeile gezeichnet. Neben den Pfeilen stehen die *Übergangsraten.*

In unserem Modell ist ein *Zustand* gegeben durch die Anzahl der Ereignisse, die bis dahin bereits eingetreten sind. Diese Zahlen stehen in den Kreisen. Wir setzen außerdem voraus, dass die Übergangsraten alle gleich sind, nämlich gleich λ.

Eine Übergangsrate λ für den Übergang vom Zustand i zum Zustand j besagt, dass die Wahrscheinlichkeit des Übergangs von i nach j in einem hinreichend kleinen Zeitintervall Δt näherungsweise gleich $\Delta t\cdot\lambda$ ist, vorausgesetzt, zu Beginn des betrachteten kleinen Zeitintervalls befindet sich das System tatsächlich im Zustand i.

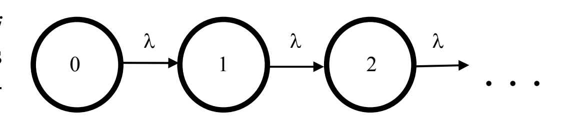


Bild 3.4 Markoff-Modell eines Poisson-Prozesses

Die Wahrscheinlichkeit des Zustands i wird mit $p_i(t)$ bezeichnet. Für den Prozess ergeben sich folgende Beziehungen für die Wahrscheinlichkeiten der Zustände:

$$p_0(t+\Delta t) = (1 - \Delta t\cdot\lambda)\cdot p_0(t),$$

$$p_{i+1}(t+\Delta t) = (1 - \Delta t\cdot\lambda)\cdot p_{i+1}(t) + \Delta t\cdot\lambda\cdot p_i(t) \text{ für } i = 0, 1, 2, ...$$

Der Grenzübergang $\Delta t\rightarrow 0$ liefert ein System von *Differentialgleichungen*:

$$\dot{p}_0(t) = -\lambda\cdot p_0(t)\,,$$

$$\dot{p}_{i+1}(t) = -\lambda\cdot p_{i+1}(t) + \lambda\cdot p_i(t) \text{ für } i = 0, 1, 2, ...$$

Anfangs hat das System noch nicht versagt. Das liefert die Anfangsbedingungen $p_0(0) = 1$, $p_1(0) = 0$, $p_2(0) = 0$, ...

Dieses *Anfangswertproblem* lässt sich sukzessive, ausgehend vom Index 0 aufsteigend, mit elementaren Methoden der Analysis lösen. Hier das Ergebnis:

$$p_i(t) = \frac{(\lambda t)^i}{i!}e^{-\lambda t} \text{ für } i = 0, 1, 2, ...$$

Die Verweilzeit im Zustand i-1, das ist die Zeit zwischen dem i-1-ten und dem i-ten Ereignis, bezeichnen wir mit T_i. Diese Zeiten heißen *Versagensabstände*, *Ausfallabstände* oder auch *Zwischenankunftszeiten*, je nachdem, ob ein Versagensprozess, ein Ausfallprozess oder ein Wartesystem modelliert wird. Der Zustand i liegt zum Zeitpunkt t genau dann vor, wenn die Bedingung $T_1 + T_2 + ... + T_i \leq t < T_1 + T_2 + ... + T_i + T_{i+1}$ erfüllt ist.

Der eingeführte Strom von Ereignissen hat exponentialverteilte *Zeiten zwischen den Ereignissen.* Er wird *Poisson-Strom* genannt. Die Übergangsrate ist im Falle des Versagensprozesses

gleich der weiter oben eingeführten Versagensrate. Weiterführende Literatur: Fisz (1976, S. 326 ff.), Kleinrock (1975, Band 1, S. 61 ff.). Nehmen wir als Beispiel eine

Garantie. Für einen Computer mit einer geschätzten konstanten Versagensrate von 6/Jahr wird per Kaufvertrag vereinbart, dass der Verkäufer den Rechner zurücknehmen muss, wenn es im ersten Monat zu mehr als 2 Versagensfällen kommt. Die Wahrscheinlichkeit dafür, dass der Verkäufer ein ausgeliefertes Gerät zurücknehmen muss, ergibt sich folgendermaßen: Die Wahrscheinlichkeit dafür, dass das Gerät keinmal, einmal oder gar zweimal versagt, ist gegeben durch $p_0(t)+ p_1(t)+ p_2(t)$. Wegen $\lambda\cdot t$=0.5 ist dieser Wert gleich $(1+0.5+(0.5)^2/2)\cdot e^{-0.5}$, also gleich 0.9856. Für den Verkäufer ergibt sich eine Wahrscheinlichkeit von weniger als zwei Prozent für die Rücknahme (genauer: 1.44 %).

Tabelle 3.1 Zuverlässigkeitsdaten von Kraftwerkskomponenten (Kuhlmann, 1981; BMFT, 1980; Shooman, 1990). Falls die Kennzahl nicht auf eine Zeiteinheit wie Stunde (h) oder Jahr (a) bezogen ist, handelt es sich um eine Versagenswahrscheinlichkeit je Anforderung.

Komponente	**Ausfallart**	**Ausfallrate bzw. Versagensrate**
Rohrleitungen	Versagen	Wöhlerlinie
Rohrleitungen	Kleines Leck	$<3_{10}$-3/a
Rohrleitungen	Großes Leck	$<1.5_{10}$-4/a
Pumpe	Betriebsversagen	2.5_{10}-5/h
Handarmatur	öffnet/schließt nicht	1.5_{10}-6/h
Regelarmatur	fährt nicht auf/zu	5_{10}-6/h
Umschaltventil	schaltet nicht um	4_{10}-2
Steuerventil	öffnet/schließt nicht	4_{10}-3
Entlastungsventil	öffnet unbeabsichtigt	2_{10}-6/h
Dichtung	Leckage	4_{10}-7/h
Schütz	Verlust der Hauptfunktion	1_{10}-6/h
Sicherung	vorzeitige Unterbrechung	1_{10}-6/h
Transformator		1.5_{10}-6/h
Elektronik-Karte		0.2_{10}-6/h ... 1.2_{10}-6/h
Mikroelektronik mit *n* Gattern		$C\cdot n^{1/2}$ mit $C = 0.4_{10}$-9/h

4 Qualitätskontrolle und Zuverlässigkeitsschätzung

Im letzten Kapitel wurden Werte für die Bauelementeausfallraten und für die Versagenswahrscheinlichkeiten von Bauelementen und Subsystemen angegeben. Wie aber kommt man zu solchen Werten? Wie verlässlich sind sie? Der erste Punkt betrifft die

- *Schätzung* des Parameters einer Verteilung und der zweite den
- *Nachweis*, dass der Parameter nicht schlechter als ein bestimmter angenommener Wert ist.

Anhand der statistischen Qualitätskontrolle werden die Grundideen der *Parameterschätzung* und des *Nachweises* eingeführt. Die Bücher von Fisz (1976), Sachs (1992), Birolini (1991) und Geiger (1998) geben Auskunft über Details.

4.1 Schätzung und Nachweis der Fehlerwahrscheinlichkeit

In der statistischen Qualitätskontrolle geht es darum, anhand einer Stichprobe des Umfangs N eine hinreichend genaue Aussage über die Fehlerwahrscheinlichkeit (auch: *Ausschusswahrscheinlichkeit*) p eines Produkts zu erhalten. Es ist also ein Schätzwert für p zu ermitteln und im Anschluss daran der Nachweis zu erbringen, dass der tatsächliche Wert - mit einer bestimmten Vertrauenswahrscheinlichkeit - nicht schlechter als ein vorgegebener Grenzwert ist.

Allgemeiner gelten die Überlegungen für N-fach wiederholte und unabhängige Tests, von denen jeder mit der bestimmten - aber unbekannten - Wahrscheinlichkeit p positiv ist. Das Ergebnis eines Tests gilt als *negativ*, wenn der Test bestanden wird, und als *positiv*, wenn er einen Fehler offenbart.

Es kann sich bei den Tests also um die Tests verschiedener Exemplare eines Produkts handeln, aber auch um den wiederholten Test eines einzigen Geräts, der nur mit einer bestimmten Wahrscheinlichkeit einen Fehler offenbart. Ein Beispiel dafür ist die Typprüfung auf elektromagnetische Verträglichkeit: Ob der Prüfling gegenüber einem bestimmten Impuls als störfest anzusehen ist, wird durch N-maliges Beaufschlagen des Prüflings mit diesem Impuls getestet. Ähnlich liegen die Dinge beim Test eines Programms mit unabhängig voneinander gewählten Eingabedatensätzen. Hier ist der Parameter p nichts anderes als die zu bestimmende Versagenswahrscheinlichkeit des Programms.

Parameterschätzung

Ordnet man jedem positiven Testergebnis den Wert 1 und dem negativen den Wert 0 zu und addiert diese Werte X_i über die gesamte Stichprobe, dann erhält man eine (N, p)-binomialverteilte Zufallsvariable $X = X_1 + X_2 + ... + X_N$. Wegen $E[X_i] = p$ kann der Parameter p mit derselben Methode wie alle Mittelwerte geschätzt werden (Abschnitt 2.2).

Der Schätzwert $m = X/N$ ist gleich der Anzahl n der positiven Testergebnisse geteilt durch den Stichprobenumfang: $m = n/N$. Außer der Unabhängigkeit der Stichprobenwerte und der gemeinsamen zugrundeliegenden Verteilung setzen wir voraus, dass der Schätzwert m für p als näherungsweise normalverteilt vorausgesetzt werden kann (s. Bild 2.2) und dass

$$s = \sqrt{(\sum_{i=1}^{N} X_i^2 - N \cdot m^2)/(N-1)} = \sqrt{N/(N-1)} \cdot \sqrt{m(1-m)}$$

ein guter Schätzwert für die Standardabweichung der X_i ist. Diese Annahme ist für $Np(1-p) \geq 9$ berechtigt (Sachs, 1992, S. 270).

Die maximale Abweichung im $t\sigma$-Intervall ergibt sich damit zu $E = ts/\sqrt{N} = t\sqrt{m(1-m)/(N-1)}$. Also gilt: $p = m \pm t\sqrt{m(1-m)/(N-1)}$ mit der Vertrauenswahrscheinlichkeit Q, wobei das zu einem bestimmten Q gehörige t wieder der Tabelle 2.1 entnommen werden kann.

Die Annahme $Np(1-p) \geq 9$ schränkt die Anwendbarkeit der Formel stark ein: Oft ist der Nachweis gefordert, dass p sehr klein ist. Man möchte also nur wenige - möglichst gar keine - positiven Testresultate sehen. Das impliziert, dass $Np<1$ ist, und das macht die Anwendung der Formel unmöglich. Im folgenden Abschnitt geht es vor allem um den Nachweis kleiner Fehler- bzw. Versagenswahrscheinlichkeiten p.

4.2 Nachweis kleiner Fehler- bzw. Versagenswahrscheinlichkeiten

Mit Hilfe einer Stichprobe soll eine Aussage über die Fehler- bzw. Versagenswahrscheinlichkeit gewonnen werden.

Wir wollen beispielsweise, dass die Fehlerwahrscheinlichkeit unterhalb einer gewissen Schranke p liegt. Diesen erwünschten Sachverhalt bezeichnen wir als *Nullhypothese.*

Dem Nachweis der Nullhypothese dient ein Test. Für diesen *Test* nehmen wir eine Stichprobe der Größe N. Der Test gilt als bestanden, wenn sich in dieser Stichprobe höchstens k Fehler (positive Testfälle) befinden. Der Test trägt die Bezeichnung $T_{k,N}$.

Auf welchen Wert müssen wir k festlegen, wenn mit dem Test $T_{k,N}$ der Nachweis der Nullhypothese gelingen soll?

Zur Beantwortung der Frage nehmen wir vorläufig einen negativen Standpunkt ein: Wir vermuten, dass die Hypothese, dass die Fehlerwahrscheinlichkeit kleiner als p ist, nicht eingehalten wird. Ganz speziell untersuchen wir die Hypothese, dass der Parameter genau gleich p ist. Diese Hypothese bezeichnen wir mit $H(p)$.

Die Hypothese $H(p)$ wollen wir als widerlegt ansehen, wenn der Test $T_{k,N}$ bestanden wird. Da bei Vergrößerung des Parameters p das Bestehen des Tests eher unwahrscheinlicher wird, bedeutet das Bestehen des Tests, dass der tatsächliche Parameter höchstwahrscheinlich kleiner als p ist. "Höchstwahrscheinlich" steht dafür, dass mit der Wahrscheinlichkeit β ein Irrtum vorliegen kann. Damit ist der geforderte Nachweis gelungen.

In der statistischen Testtheorie nennt man das fälschliche Beibehalten der Nullypothese einen *Fehler zweiter Art*. Art: Ein Fehler zweiter Art liegt demnach vor, wenn der Test bestanden wird, obwohl die Alternativhypothese gilt. Der Fehler zweiter Art ist also ein Annahmefehler. (Demgegenüber bezeichnet man das fälschliche Ablehnen der Nullhypothese im Rahmen eines Tests, den Ablehnungsfehler also, als Fehler erster Art.)

Die Wahrscheinlichkeit β dafür, dass die Nullhypothese beibehalten wird, obwohl die Alternativhypothese $H(p)$ gilt, wollen wir *Annahmewahrscheinlichkeit* nennen.

Unter Zugrundelegung der Hypothese $H(p)$ fragen wir also nach der Wahrscheinlichkeit β, mit der der Test $T_{k,N}$ bestanden wird. Diese Wahrscheinlichkeit ist die *Operationscharakteristik* des Tests $T_{k,N}$ (auch: Testcharakteristik). Sie wird mit $L_{k,N}(p)$:

$$L_{k,N}(p) = P(n \leq k \mid H(p)).$$

$P(A|B)$ steht für „Wahrscheinlichkeit von A unter der Bedingung B“.

Die Testcharakteristik ist definitionsgemäß gleich der *Annahmewahrscheinlichkeit.* Das ist die Wahrscheinlichkeit β dafür, dass der Test bestanden wird, obwohl die Bedingung, dass die Fehlerwahrscheinlichkeit unterhalb p liegt, gerade eben nicht eingehalten wird. Anders ausgedrückt: Mit der Annahmewahrscheinlichkeit wird die Nullhypothese akzeptiert, obwohl $H(p)$ gilt. Die Operationscharakteristik des beschriebenen Tests ergibt sich zu

$$L_{k,N}(p) = P(n \leq k \mid H(p)) = \sum_{n=0}^{k} \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \; .$$

Bild 4.1 zeigt Operationscharakteristiken für $N = 10$. Zwischen den Operationscharakteristiken besteht die Beziehung $L_{k,N}(p) + L_{N-k-1,N}(1-p) = 1$. Es folgen ein paar Beispiele.

Testplanung. Nachzuweisen ist eine Versagens- oder Fehlerwahrscheinlichkeit von höchstens 0.2 mit einer Stichprobe des Umfangs $N = 10$. Eine Annahmewahrscheinlichkeit von 10 % wird zugelassen. Die Testcharakteristik $L_{0,10}(p)$ zeigt, dass der Test $T_{0,10}$ für den Nachweis geeignet ist, d. h.: unter 10 Testfällen darf kein einziger positiv sein.

Wir fragen nun, wie groß eine Stichprobe sein muss, um - unter Zugrundelegung einer bestimmten Annahmewahrscheinlichkeit β - eine Fehler- oder Versagenswahrscheinlichkeit kleiner p nachweisen zu können. Wir setzen voraus, dass der Nachweis für ein extrem kleines p zu führen ist. Der Test soll kein einziges positives Ergebnis bringen. Die zugehörige Testcharakteristik bei der Stichprobengröße N ist demnach gleich $L_{0,N}(p) = (1-p)^N$.

Daraus ergibt sich $\beta = (1-p)^N$. Oder: $N = \ln(\beta)/\ln(1-p) \approx -\ln(\beta)/p$. Bei einer Annahmewahrscheinlichkeit von $\beta = 5$ % ist $-\ln(\beta) \approx 3$. Daraus folgt: Man braucht eine Stichprobengröße von wenigstens $N = 3/p$, wenn man eine Fehlerwahrscheinlichkeit von höchstens p mit einer Annahmewahrscheinlichkeit von 5 % nachweisen will.

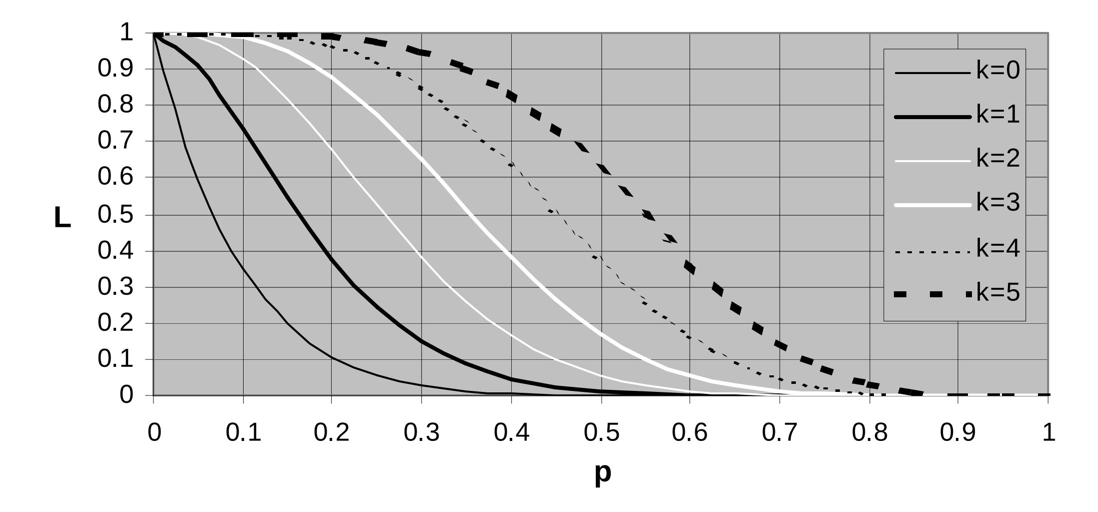


Bild 4.1 Operationscharakteristiken $L_{k,N}(p)$ für die Stichprobengröße $N = 10$

Wareneingangsprüfung. Eine Lieferung elektronischer Schaltkreise ist zu begutachten. Der Test eines jeden Schaltkreises ist sehr aufwendig, so dass Lieferant und Abnehmer sich auf eine Stichprobengröße von $N = 10$ geeinigt haben. Der Abnehmer soll zu 80 % sicher sein können, dass höchstens 30 % der Geräte fehlerhaft sind. Die Operationscharakteristik $L_{1,10}(p)$

erfüllt die Bedingung. Der Kunde akzeptiert die Sendung, wenn höchstens eins von zehn getesteten Geräten fehlerhaft ist.

Software-Test. Ein zyklisch arbeitendes Programm mit einer Zyklusdauer im Millisekundenbereich soll eine Versagensrate von höchstens 10^{-4}/h besitzen. Das ist eine moderate Anforderung an die Zuverlässigkeit. Diese Versagensrate führt auf längere Sicht zu etwa je einem Versagensfall jährlich. Der Nachweis soll mit einer Annahmewahrscheinlichkeit von 5 % geführt werden. Aus der Versagensrate lässt sich (bei bekannter Zyklusdauer) die Versagenswahrscheinlichkeit p je Zyklus ermitteln. Für den Test wird die Operationscharakteristik $L_{0,N}(p)$ zu Grunde gelegt. Wegen $N = 3/p$ ergibt sich eine minimale Testdauer von $3\cdot10^4$ Stunden. Also: Auch bescheidene Zuverlässigkeitsanforderungen laufen auf einen horrenden Testaufwand hinaus. Von sicherheitsrelevanter Software wird eine Versagensrate von unter 10^{-7}/h gefordert. Die Autoren Ricky W. Butler und George B. Finelli (1993) kommen wie andere auch zum Schluss, dass die Zuverlässigkeit sicherheitsrelevanter Software mit statistischen Methoden nicht ermittelt werden kann.

4.3 Bayes-Schätzung kontra Testtheorie

Eine fragwürdige Formel. Zum Nachweis kleiner Fehlerwahrscheinlichkeiten wird ein Test mit der Stichprobengröße N durchgeführt. Der Test ist negativ (gilt als bestanden), wenn in der Stichprobe kein Fehler auftritt. Bei negativem Test gilt die Hypothese einer kleinen Fehlerwahrscheinlichkeit als bestätigt.

Die (als relativ klein angesetzte) Annahmewahrscheinlichkeit β, die Stichprobengröße N und die nachzuweisende Obergrenze p für die Fehlerwahrscheinlichkeit hängen folgendermaßen zusammen: $\beta = (1\text{-}p)^N$.

Diesen Zusammenhang liest man so: Der Test wird von einem Los mit der gerade eben nicht mehr tolerierten Fehlerwahrscheinlichkeit p mit der Annahmewahrscheinlichkeit β bestanden.

Zuweilen wird der Zusammenhang aber so verstanden (Ehrenberger, 2002, S.117): Bei negativem Testergebnis ist die tatsächliche Fehlerwahrscheinlichkeit w des Loses mit der Wahrscheinlichkeit 1-β kleiner als p. Diese Lesart kommt in der *Formel* $P(w < p) = 1\text{-}(1\text{-}p)^N = 1\text{-}\beta$ zum Ausdruck. Diese Formel wird damit begründet, dass man ja den ungewissen Parameter w auch als Zufallsvariable ansehen könne, für die man Wahrscheinlichkeitsaussagen machen könne.

Die Fehlerwahrscheinlichkeit w kann man durchaus als Zufallsvariable auffassen. Dazu muss man sich verschiedene Realisierungen der Fehlerwahrscheinlichkeit zumindest vorstellen können. Das ist nicht allzu schwer: Nehmen wir als Beispiel die Herstellung eines bestimmten Produkts. Aufgrund der Unvollkommenheit des Produktionsprozesses liefert die Fabrik hin und wieder fehlerhafte Produkte aus. Die Fehlerwahrscheinlichkeit möge von Los zu Los schwanken. Wir betrachten also die Fehlerwahrscheinlichkeit w eines Loses als Zufallsvariable. Ähnliche Überlegungen lassen sich für die Versagenswahrscheinlichkeiten von Programmen anstellen.

So weit, so gut. Aber das heißt noch nicht, dass die Formel stimmt. Die statistische Testtheorie, in deren Rahmen sie vermeintlich hergeleitet wurde, gibt diese Formel nicht her. Die Formel taucht im Rahmen der Testtheorie folglich auch nirgends auf, so schön sie ist. Stattdessen zieht man sich in der statistischen Testtheorie auf ziemlich sperrige Aussagen zurück, beispielsweise darauf, dass ein „Test von einem Los mit der gerade eben nicht mehr tolerierten Fehlerwahrscheinlichkeit p mit der Annahmewahrscheinlichkeit β bestanden“ wird.

Der Widerspruch. Ein einfacher Grenzfall macht den Widerspruch deutlich und zeigt, in welche Falle man mit der fragwürdigen Formel tappt.

Nehmen wir einmal an, dass die Fabrik so arbeitet, dass die Zufallsvariable w – die Fehlerwahrscheinlichkeit – auf dem Intervall [0, 1) gleichverteilt ist. Wir halten also alle möglichen Fehlerwahrscheinlichkeiten für gleich wahrscheinlich. Das ist Ausdruck unserer totalen Unwissenheit.

Für ein bestimmtes Los wollen wir die Gültigkeit der Hypothese $w < p$ nachweisen. Ich nehme jetzt als Extremfall eine leere Stichprobe: $N = 0$. Dieser Test wird auf jeden Fall bestanden. Konsequenterweise ergibt sich eine Irrtumswahrscheinlichkeit von $\beta = 1$. Nach der fragwürdigen Formel wäre $P(w < p) = 0$. Die Wahrscheinlichkeit der Hypothese ist – aufgrund der Gleichverteilungsannahme – aber gegeben durch $P(w < p) = p$.

Hier wurde im Bayes-Stil argumentiert: Es wurde die Frage gestellt, wie sich eine Anfangsschätzung aufgrund eines Tests verbessern lässt. Und man sieht, worauf es ankommt: Die Bayes-Formel sagt etwas über die *Verbesserung von Schätzungen*. Und dafür braucht man eben eine Anfangsschätzung, ein Apriori. Im Beispiel ist es die Gleichverteilungsannahme nach dem Indifferenzprinzip.

Gerd Gigerenzer (2004) schreibt über den hier deutlich werdenden Widerspruch: „Viele glauben irrtümlicherweise, ein signifikantes Ergebnis … würde die Wahrscheinlichkeit angeben, dass die Nullhypothese richtig sei oder dass die Alternativhypothese falsch sei. Anders als die Regel von Bayes kann jedoch ein Nullhypothesentest keine Wahrscheinlichkeit für Hypothesen erbringen, lediglich eine Wahrscheinlichkeit für die Daten unter der Annahme, dass die Nullhypothese wahr ist“.

Das Vergessen der Anfangsschätzungen ist eine gern genommene Denkfalle. Weitere Beispiele sind die Harvard-Medical-School-Studie und mein Beispiel aus der Konsumforschung. Und sogar Charles Perrow ist in seinem Buch „Normal Accidents“ in diese Falle getappt (Grams, 2006).

Die Bayes-Schätzung. Wir benötigen eine Anfangsbeschreibung der Produktionsverhältnisse, denn ansonsten könnten wir gar keine Wahrscheinlichkeiten für die Gültigkeit der Hypothese oder das Auftreten von bestimmten Stichprobenwerten angeben. Unsere Grundannahme ist, dass die Fabrik Lose liefert, deren Fehlerwahrscheinlichkeiten w im Intervall $[a, b)$ liegen und dort gleich verteilt sind. Für $a = 0$ und $b = 1$ haben wir den oben beschriebenen Sonderfall der totalen Unwissenheit. Die Hypothese H sei $w < p$. Für die folgende Rechnung wird $a \leq p \leq b$ vorausgesetzt. Die Beobachtung B ist die fehlerfreie Stichprobe des Umfangs N, der negative Test also.

Die A-priori-Wahrscheinlichkeit der Hypothese ist gleich $P(H) = P(w < p) = \frac{p-a}{b-a}$. Daraus ergeben sich die Verteilungsfunktion F und die Verteilungsdichte f der Zufallsvariablen w im Intervall $[a, b]$ zu

$$F(x) = \frac{x-a}{b-a} \quad \text{und} \quad f(x) = \frac{1}{b-a}.$$

Die Wahrscheinlichkeit einer fehlerfreien Stichprobe der Größe N ist unter den gegebenen Produktionsbedingungen gleich

$$P(B) = \int_a^b (1-x)^N \cdot \frac{1}{b-a} \cdot \mathrm{d}x = \frac{(1-a)^{N+1} - (1-b)^{N+1}}{(b-a)\cdot(N+1)}.$$

Bei Gültigkeit der Hypothese ist in den Formeln für F und f jeweils das b durch das p zu ersetzen. Die Wahrscheinlichkeit der Beobachtung ist dann gleich

$$P(B \mid H) = \int_a^p (1-x)^N \cdot \frac{1}{p-a} \cdot \mathrm{d}x = \frac{(1-a)^{N+1} - (1-p)^{N+1}}{(p-a)\cdot(N+1)} .$$

Die gesuchte A-posteriori-Wahrscheinlichkeit der Hypothese ist damit gegeben durch

$$P(H \mid B) = \frac{P(B \mid H) \cdot P(H)}{P(B)} = \frac{(1-a)^{N+1} - (1-p)^{N+1}}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Die nächste Verbesserung. Wir setzen nun die gemachte Beobachtung B voraus und haben für die Hypothese *H,* nämlich für $w < p$, die Wahrscheinlichkeit

$$P(H) = P(w < p) = \frac{(1-a)^{N+1} - (1-p)^{N+1}}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Das bedeutet nichts anderes, als dass die Fehlerwahrscheinlichkeit w jetzt die Verteilungsfunktion F hat, die auf dem Intervall $[a, b]$ gegeben ist durch

$$F(x) = \frac{(1-a)^{N+1} - (1-x)^{N+1}}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Die zugehörige Verteilungsdichte ist

$$f(x) = F'(x) = \frac{(N+1)\cdot(1-x)^N}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Nun wird ein Test mit der Stichprobengröße M durchgeführt. Er sei negativ. Das ist unsere neue Beobachtung B. Ihre Wahrscheinlichkeit ist bei der gegebenen Verteilung gleich

$$P(B) = \int_a^b (1-x)^M \cdot f(x) \cdot \mathrm{d}x = \frac{N+1}{M+N+1} \cdot \frac{(1-a)^{M+N+1} - (1-b)^{M+N+1}}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Die Wahrscheinlichkeit der Beobachtung unter der Bedingung, dass die Hypothese $w < p$ gilt, ist gegeben durch

$$P(B \mid H) = \int_a^p (1-x)^M \cdot \frac{f(x)}{F(p)} \cdot \mathrm{d}x = \frac{1}{F(p)} \cdot \frac{N+1}{M+N+1} \cdot \frac{(1-a)^{N+N+1} - (1-p)^{M+N+1}}{(1-a)^{N+1} - (1-b)^{N+1}} .$$

Für die A-posteriori-Wahrscheinlichkeit der Hypothese $w < p$ gilt dann

$$P(H \mid B) = \frac{P(B \mid H) \cdot P(H)}{P(B)} = \frac{(1-a)^{M+N+1} - (1-p)^{M+N+1}}{(1-a)^{M+N+1} - (1-b)^{M+N+1}} .$$

Dieselbe A-posteriori-Wahrscheinlichkeit hätte sich ergeben, wenn man nicht in zwei Stufen vorgegangen wäre, sondern wenn man gleich eine Stichprobe des Umfangs M+N gewählt hätte.

Korrektur der Formel. Wir setzen totale Unwissenheit voraus. Unter der Bedingung, dass ein Test der Stichprobengröße N bestanden wird, ergibt sich die Formel $P(w < p) = 1-(1-p)^{N+1}$. Sie entsteht aus der infrage gestellten Formel, indem man das N durch N+1 ersetzt.

Allerdings ist diese Formel nicht in der statistischen Testtheorie angesiedelt, sondern in der „Welt von Bayes“. Und außerdem kommen wir um eine zusätzliche Annahme bezüglich des Apriori nicht herum.

Wir benutzen das *Indifferenzprinzip*. So hat der berühmte Volkswirtschaftler und Autor des Buches *A Treatise on Probability* John Maynard Keynes das „Prinzip vom mangelnden zureichenden Grunde“ genannt: „Wenn keine Gründe dafür bekannt sind, um eines von verschiedenen möglichen Ereignissen zu begünstigen, dann sind die Ereignisse als gleich wahrscheinlich anzusehen“ (zitiert nach Carnap/Stegmüller, 1958, S. 3).

Das Indifferenzprinzip darf man nicht auf die leichte Schulter nehmen: Im Grunde ist es eine Annahme über etwas, wovon man überhaupt nichts weiß. Die Annahme bleibt letztlich unbegründet.

Die Testtheorie kommt ohne solche Annahmen aus. Das ist ihre entscheidende Stärke gegenüber der Argumentation auf der Linie der Bayes-Formel. Sicherheits- und Zuverlässigkeitsnachweise wird man folglich möglichst im Rahmen der Testtheorie durchführen, selbst wenn die Zuverlässigkeitsaussagen dort sperriger zu formulieren und dementsprechend auch etwas schwerer zu verstehen sind.

4.4 Zuverlässigkeitsschätzung

Wir betrachten ein System, dessen Ausfallprozess bzw. Versagensprozess als Poisson-Prozess beschrieben werden kann. Die Abschätzung der Zuverlässigkeit des Systems geschieht auf der Basis von n bisher beobachteten Versagensabständen $t_1, t_2, ..., t_n$. Diese Werte sind Realisierungen der Zufallsvariablen $T_1, T_2, ..., T_n$. Die Zeiten zwischen aufeinanderfolgenden Versagensfällen $T_1, T_2,...$ sind exponentialverteilt mit dem Parameter λ (Versagensrate). Die Erwartungswerte der T_i sind alle gleich: $E[T_i] = 1/\lambda$.

Der Erwartungswert der Versagensabstände wird durch den arithmetischen Mittelwert der bisher beobachteten Versagensabstände abgeschätzt: $(t_1 + t_2 + ... + t_n)/n$. Daraus ergibt sich für die Versagensrate λ der Schätzwert $\lambda^* = n/(t_1+t_2+ ...+t_n)$. Die Genauigkeit des Schätzwerts λ^* lässt sich mit Hilfe der Poisson-Verteilung (Abschnitt 3.6) abschätzen.

Wir setzen eine bestimmte Versagensrate λ voraus. Die Zeit bis zum n-ten Versagen ist $T = T_1 + T_2 + ... + T_n$. Die Verteilungsfunktion der Zeit bis zum n-ten Versagen ist $P(T < t)$. Mit den Wahrscheinlichkeiten $p_i(t)$ für die Zustände i eines Poisson-Prozesses zum Zeitpunkt t ergibt sich diese Wahrscheinlichkeit zu $p_n(t) + p_{n+1}(t) + ...$ bzw. zu $1- (p_0(t) + p_1(t) + ... + p_{n-1}(t))$. Mit den Ergebnissen des Abschnitts 3.6 folgt daraus

$$P(T < t) = 1 - \sum_{i=0}^{n-1} \frac{(\lambda \cdot t)^i}{i!} e^{-\lambda \cdot t} \ .$$

In der Formel für die Verteilungsfunktion kommen λ und t nur als Produkt vor. Deshalb definieren wir die normierte Verteilungsfunktion $L_n(x)$ folgendermaßen

$$L_n(x) = 1 - \sum_{i=0}^{n-1} \frac{(x)^i}{i!} e^{-x} \ .$$

Der Summenausdruck ist einem leichter zu handhabenden Integralausdruck äquivalent:

$$L_n(x) = \int_0^x \frac{u^{n-1}}{(n-1)!} \cdot e^{-u} \mathrm{d}u \ .$$

Verteilungsfunktion und normierte Verteilungsfunktion hängen folgendermaßen zusammen

$$P(T < t) = L_n(\lambda \cdot t).$$

Berechnung des Vertrauensintervalls einer Zuverlässigkeitsschätzung. Wir setzen das Konfidenzniveau auf α, zum Beispiel: $\alpha = 5\%$, und bestimmen die Werte a und b so, dass die Gleichungen $L_n(a) = \alpha$ und $L_n(b) = 1-\alpha$ erfüllt sind. Sei t^* die Zeit bis zum tatsächlich beobachteten n-ten Versagen: $t^* = t_1 + t_2 + \ldots + t_n$. Unter der Annahme, dass die Versagensrate gleich λ ist, liegt der ermittelte Wert t^* mit der Wahrscheinlichkeit α unterhalb von a/λ; und mit der Wahrscheinlichkeit $1-\alpha$ liegt der Wert t^* unterhalb von b/λ. Also liegt t^* mit der Wahrscheinlichkeit $1-2\alpha$ im Intervall $[a/\lambda, b/\lambda)$.

Es ist $\lambda^* = n/(t_1+t_2+ \ldots +t_n) = n/t^*$. Die Bedingungen $a/\lambda \leq t^*$, $a/t^* \leq \lambda$ und $\frac{a}{n}\lambda^* \leq \lambda$ sind allesamt äquivalent. Dasselbe gilt für die Bedingungen $\lambda < b/t^*$, $t^* < b/\lambda$ und $\lambda < \frac{b}{n}\lambda^*$. Wir können also zu $(1-\alpha)\cdot 100\%$ sicher sein, dass das unbekannte λ durch die zufällige Grenze $\frac{a}{n}\lambda^*$ nach unten begrenzt ist. Analog gilt, dass λ mit der Wahrscheinlichkeit $(1-\alpha)\cdot 100\%$ durch die zufällige Grenze $\frac{b}{n}\lambda^*$ nach oben begrenzt ist. Das *Vertrauensintervall* für λ zur Aussagesicherheit $(1-2\alpha)\cdot 100\%$ ist $[\frac{a}{n}\lambda^*, \frac{b}{n}\lambda^*)$.

Hier gilt dasselbe wie im Abschnitt 2.4: Das Vertrauensintervall ist ein Zufallsergebnis, das den festen aber unbekannten Wert λ höchstwahrscheinlich einschließt. Dabei steht höchstwahrscheinlich für die Wahrscheinlichkeit $(1-2\alpha)\cdot 100\%$.

Bild 4.2 zeigt die normierten Verteilungsfunktionen der Zeiten bis zum fünften und bis zum zehnten Versagen. Wir lesen daraus ab, dass $L_5(2) \approx 5\%$ ist, und dass $L_5(9) \approx 95\%$ ist. Daraus ergibt sich, dass wir zu 90% sicher sein können, dass λ die Bedingung $0.4\lambda^* \leq \lambda \leq 1.8\lambda^*$ erfüllt. Legt man für die Abschätzung 10 Versagensfälle zugrunde, ergeben sich die folgenden Ungleichungen: $0.54\lambda^* \leq \lambda \leq 1.57\lambda^*$.

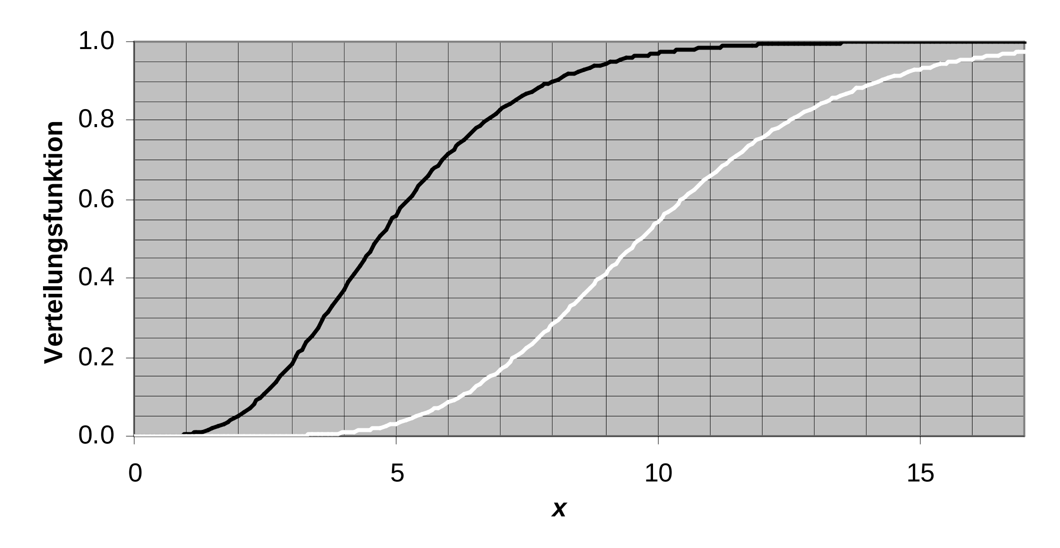


Bild 4.2 Die normierten Verteilungsfunktionen $L_5(x)$ (dunkle Kurve) und $L_{10}(x)$ (helle Kurve)

5 Diagnostizierbarkeit und Fehlertoleranz

Unter *Fehlertoleranz* wird die Fähigkeit eines Systems verstanden, auch mit einer begrenzten Zahl fehlerhafter Subsysteme seine Spezifikation zu erfüllen. So etwas geht im Allgemeinen nur, wenn mehr technische Mittel bereitstehen, als für die vorgesehene Funktion eigentlich notwendig sind. Dieser Mehraufwand gegenüber dem minimal erforderlichen heißt *Redundanz*.

Um zu verhindern, dass die Fehleranfälligkeit in allen Subsystemen gleich ist, werden diese oft mit verschiedenartigen technischen Mitteln realisiert. Unter *Diversität* versteht man solcherart ungleichartige technische Mittel zur Erreichung nützlicher Redundanz (z.B. andere physikalische Prinzipien, andere Lösungswege der gleichen Aufgabe usw.).

Die *M*-aus-*N*-Systeme sind in diesem Sinne redundant. Außerdem lassen sich die Subsysteme diversitär realisieren. Diese Systeme tolerieren Fehler in Subsystemen. Wie aber steht es mit dem Vergleicher? Soll das gesamte System fehlertolerant sein, muss der Vergleicher selber fehlertolerant aufgebaut werden. Eine solche Lösung findet man in der Sicherheitstechnik.

Aber es geht auch anders. Gerade bei hochkomplexen Anlagen, in denen die parallelredundanten Teile aus Programmen oder ganzen Rechnern bestehen, kann man die Vergleichsfunktionen dezentralisieren, indem man sie auf die Subsysteme verteilt. Allgemein kann man die Subsysteme eines komplexen Systems mit der Fähigkeit ausstatten, einander zu diagnostizieren.

Die Theorie der *Diagnostizierbarkeit* bietet eine Reihe auch für die Praxis hochinteressante Ergebnisse. Vor allem sagt sie, welchen Aufwand man zur Erreichung einer bestimmten geforderten Diagnosefähigkeit wenigstens treiben muss. Und darüber hinaus verrät sie auch noch effiziente Lösungen (Preparata, Metze, Chien, 1967; Görke, 1989, S. 109 ff.).

Modell: Die Subsysteme sind die Knoten eines gerichteten Graphen. Ein Pfeil vom Knoten i zum Knoten j des Graphen entspricht einer Diagnose des Subsystems j durch das Subsystem i. Das Diagnoseergebnis (auch: Testergebnis) wird durch den Wert f_{ij} dargestellt. Er ist gleich 0, wenn Knoten i den Knoten j für intakt hält, und gleich 1, wenn ein Defekt angezeigt wird. Wichtig ist nun, dass defekte Einheiten unzuverlässige Diagnoseergebnisse liefern. Eine defekte Einheit kann als Testergebnis 0 oder 1 melden und das besagt nichts über die diagnostizierte Einheit; diese kann defekt sein oder auch nicht. Eine intakte Einheit liefert dahingegen immer korrekte Diagnoseergebnisse. Alle Möglichkeiten sind in der Tabelle 5.1 zusammengefasst.

Tabelle 5.1 Diagnoseergebnisse

i	j	f_{ij}
intakt	intakt	0
intakt	defekt	1
defekt	intakt	0 oder 1
defekt	defekt	0 oder 1

Definition. Ein System heißt (einschrittig) *t-diagnostizierbar* (one-step *t*-fault diagnosable), wenn die Gesamtheit der Testergebnisse es erlaubt, sämtliche defekte Knoten zu identifizieren, solange höchstens t Einheiten defekt sind.

Satz 1. Ein t-diagnostizierbares System besteht aus wenigstens $2t+1$ Knoten.

Beweis durch Widerlegung des Gegenteils: Wir nehmen an, dass es ein t-diagnostizierbares System mit höchstens $2t$ Knoten gibt. Wir teilen die Knoten des Systems in zwei Teilmengen derart auf, dass jede der Teilmengen höchstens t Knoten enthält. Wir setzen nun voraus, dass alle Diagnoseergebnisse zwischen Knoten derselben Teilmenge gleich null sind. Jedes Diagnoseergebnis, bei dem der testende Knoten und der getestete Knoten verschiedenen Teilmengen angehören, möge den Wert eins haben. Kurz: Alle Testergebnisse innerhalb derselben Menge sind negativ; grenzüberschreitende Testergebnisse sind positiv. Ein solches Testergebnis ist verträglich mit der Annahme, dass sämtliche Knoten einer der Teilmengen defekt sind, und die der anderen allesamt intakt. Da die Anzahl der defekten Knoten den Wert t nicht übersteigt,

müsste sich in dieser Situation herausfinden lassen, welche Knoten die defekten sind. Wegen der Symmetrie der Testergebnisse ist es aber nicht möglich, herauszufinden, in welcher der beiden Teilmengen die defekten Knoten stecken. Das steht im Widerspruch zur t-Diagnostizierbarkeit. Damit ist die Annahme widerlegt, dass es ein t-diagnostizierbares System mit weniger als $2t$+1 Knoten gibt.

Satz 2. In jedem t-diagnostizierbaren System wird jeder Knoten von wenigstens t anderen getestet.

Beweis durch Widerlegung des Gegenteils: Wir nehmen an, dass es einen Knoten k gibt, der von weniger als t anderen Knoten getestet wird. Diese testenden Knoten mögen alle defekt sein. Auf ihr Diagnoseergebnis ist also kein Verlass. Es lässt sich nicht feststellen, ob auch der Knoten k defekt ist oder nicht. Denn: Wenn er ein Testergebnis wie ein intakter Knoten liefert (und das ist ja auch einem defekten Knoten zugestanden), dann liefert das Diagnoseergebnis insgesamt keine eindeutige Aussage über den Zustand des Knotens k. Diese Mehrdeutigkeit steht im Widerspruch zur t-Diagnostizierbarkeit.

Zentral in der Theorie der Diagnostizierbarkeit ist die Umkehrung der Sätze 1 und 2: Es gibt optimale t-diagnostizierbare Systeme, also Systeme, deren Aufwand gemessen an den obigen Sätzen minimal ist. Genau das besagt der

Satz 3. Zu jedem t gibt es ein t-diagnostizierbares System aus $n = 2t+1$ Knoten, von denen jeder von jeweils t anderen Knoten diagnostiziert wird.

Den Beweis stellen wir zurück. Wir führen zunächst Systeme mit einer bestimmten Struktur ein, nämlich die t-Diagnosezyklen. Diese bestehen aus $n = 2t+1$ Knoten, und jeder Knoten wird von genau t anderen Knoten diagnostiziert. Dann wird gezeigt, dass die t-Diagnosezyklen t-diagnostizierbar sind (Satz 4). Da es für jedes t einen t-Diagnosezyklus gibt, ist dann auch der Satz 3 auf konstruktivem Weg bewiesen.

Definition. Der *t-Diagnosezyklus* besteht aus $n = 2t+1$ Knoten, die von 0 bis n-1 durchnumeriert werden. Jeder Knoten in diesem Zyklus diagnostiziert die t auf ihn folgenden Knoten. Das heißt, der Knoten mit der Nummer i diagnostiziert den Knoten mit der Nummer j genau dann, wenn $j\text{-}i$ MOD $n \in \{1, 2, ..., t\}$. Bild 5.1 zeigt den 1-Diagnosezyklus und den 2-Diagnosezyklus.

Satz 4. Die t-Diagnosezyklen sind t-diagnostizierbar.

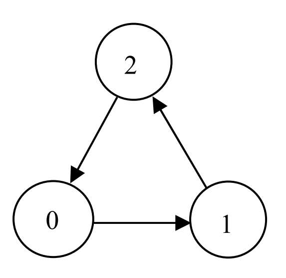
Beweis: Wir nehmen an, dass maximal t Knoten eines t-Diagnosezyklus defekt sind. Wir zeigen nun, dass das Testergebnis die defekten Knoten zu identifizieren gestattet. Zuerst wird angenommen, dass es in dem Diagnosegraphen eine gerichtete Schleife mit mehr als t Knoten gibt, bei der alle Testergebnisse der Schleife gleich 0 sind (0-Schleife). Die Knoten einer 0-Schleife müssen entweder alle defekt oder alle intakt sein. Da die betrachtete 0-Schleife mehr als t Knoten enthält, müssen sie alle intakt sein. Auf die Diagnoseergebnisse dieser Knoten kann man sich verlassen. Von den restlichen Knoten kann keiner weiter als t Knoten von einem intakten weg sein (in positiver Umlaufrichtung gezählt). Jeder dieser Knoten wird also von einem der 0-Schleife diagnostiziert. Also sind die defekten Knoten eindeutig erkennbar.

Falls es keine 0-Schleife mit mehr als t Knoten gibt, müssen sämtliche defekten Knoten im Diagnosering unmittelbar hintereinander liegen. Ansonsten gäbe es von jedem intakten Knoten zu jedem im Diagnosring folgenden intakten Knoten eine Diagnoseverbindung mit dem Wert 0, und man könnte eine 0-Schleife über sämtliche intakten Knoten bilden. Und deren Zahl ist größer als t.

Daraus folgt, dass es eine Knotennummer j gibt, so dass dieser Knoten und seine t direkten Vorgänger im Diagnosering intakt sind. Ist j der letzte intakte Knoten, dann ist sein Diagnoseergebnis für den unmittelbar folgenden Knoten gleich 1. Es ergibt sich im Zyklus also eine Folge von t Nullen und einer 1. Im Falle $t = 2$ also 001. Umgekehrt gilt: Wenn man dem Zyklus folgend auf wenigstens t Nullen trifft, denen eine Eins folgt, dann muss der letzte der Kno-

ten, also derjenige, der das Diagnoseergebnis eins liefert, intakt sein. Ansonsten wären aufgrund der irrigen Diagnoseergebnisse auch seine t Vorgänger defekt. Und das übersteigt die Anzahl der angenommenen Defekte. Damit ist der Knoten als intakt identifiziert, der alle defekten zuverlässig diagnostiziert.

Insgesamt ist damit der Beweis erbracht: Das Diagnoseergebnis gestattet jedenfalls die Identifizierung der defekten Einheiten.



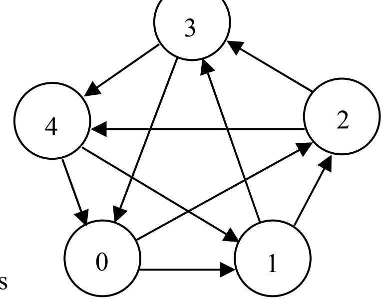


Bild 5.1 Der 1- und der 2-Diagnosezyklus

Ein Logikrätsel. Auf einer Insel leben zwei Typen von Menschen. Die eine Sorte sagt immer die Wahrheit. Die anderen Menschen lügen manchmal - aber nicht immer. Glücklicherweise gibt es nur zwei dieser "unzuverlässigen" Lügner auf der Insel. Ein Besucher trifft die Inselbewohner Albert, Bertram, Conrad, Detlef und Erich. Er will eine verlässliche Auskunft erhalten, weiß aber nicht, an wen er sich wenden soll. Ihm ist die Situation auf der Insel nämlich sehr wohl bekannt. Nur kennt er - anders als die Inselbewohner - die unzuverlässigen Lügner nicht. Also entschließt er sich, die fünf übereinander auszufragen. Hier sind die Antworten der Inselbewohner.

- Albert: "Bertram lügt manchmal, aber Conrad lügt nie".
- Bertram: "Conrad lügt manchmal, aber Detlef lügt nie".
- Conrad: "Detlef lügt manchmal, Erich nie".
- Detlef: "Erich und Albert lügen beide manchmal".
- Erich: "Albert lügt nie, Bertram manchmal".

Auf wen kann sich der Besucher verlassen?

Lösung: Man kann das Rätsel lösen, indem man alle Möglichkeiten durchspielt. Man kann die Sache als Übung in Aussagenlogik auffassen. Und man kann die Theorie der Diagnostizierbarkeit zu Hilfe nehmen. Letzteres geht meines Erachtens am schnellsten. Es geht so: Wir kürzen die Namen mit den ersten Buchstaben ab; *A* steht demnach für Albert, *B* für Bertram, usw. Zunächst stellt man fest, dass *A-B-C-D-E* ein 2-Diagnosezyklus ist. *A* diagnostiziert *B* und *C; B* diagnostiziert *C* und *D,* usw. Wir wissen, dass höchstens zwei der Personen unzuverlässige Lügner sein können. Der Beweis des Satzes 4 legt nahe, zunächst nach einer 0-Schleife mit wenigstens drei Knoten (Personen) Ausschau zu halten. Tatsächlich ist *A-C-E-A* eine Schleife mit lauter negativen Diagnoseergebnissen. Daraus folgt: Albert, Conrad und Erich lügen nie. Ansonsten hätten ja alle drei gelogen, und das kann nicht sein. Die Meinung dieser drei Aufrechten über die anderen sind wenig schmeichelhaft. Bertram und Detlef sind also tatsächlich die beiden unzuverlässigen Lügner, die es auf der Insel gibt.

6 Sicherheitstechnik

Traditionell ist die Sicherheitstechnik konservativ: Man vertraut bewährten Lösungen, Techniken und Verfahren. Da von der Technik Menschenleben abhängen, gibt es eine durchaus angemessene Scheu vor Neuem und den damit verbundenen ungewissen Gefahren.

Das Neue nicht zu wagen, bringt aber auch Gefahren mit sich. Nehmen wir als Beispiel die Wasserstoffwirtschaft. Wasserstoff als Energieträger bietet die Chance, unsere bedrohliche Abhängigkeit von den erschöpflichen Energiequellen zu mindern und die direkte Nutzung von Sonnenenergie voranzubringen. Der Umgang mit Wasserstoff ist aber eine heikle Sache. Der großtechnische Transport und die Speicherung von Wasserstoff stellen die Sicherheitstechnik vor große Herausforderungen.

Die Kernfrage lautet, wie wir die Erfahrungen und das Wissen, das in den bewährten Lösungen steckt, auf das Neue übertragen können. Mehrere Wege gibt es. Ich greife zwei extreme Vorgehensweisen heraus, die ich durch die Attribute *qualitativ* und *quantitativ* kennzeichnen will. In Reinkultur wird man sie nirgends finden. Aber: Polarisierung schafft Klarheit.

Im Zentrum der *qualitativen Sicherheitstechnik* steht die Klassifizierung von Maschinen und Komponenten hinsichtlich der von ihnen ausgehenden oder zu verhindernden Gefahren (Abschnitt 6.1). Die Gefahrenanalyse und die Festlegung der sicherheitsgerichteten technischen und organisatorischen Maßnahmen geschieht auf der Basis von Regelwerken, Checklisten und Formularen, wie beispielsweise bei der Ausfalleffektanalyse (Abschnitt 6.3). Die Entscheidungen werden durch interdisziplinär zusammengesetzte Arbeitsgruppen nach dem Konsensprinzip getroffen. Vorherrschend ist die deterministische Betrachtungsweise. Die behandelten Fragen haben die Grundformen „Was passiert, wenn... ?“ und „Wie kann es geschehen, dass... ?“ Sicherheit ist eine soziale Konstruktion in Gestalt einer Spezifikation.

Bei der *quantitativen Sicherheitstechnik* ist der Begriff des Risikos zentral (Kapitel 12). Sicherheit ist gegeben, wenn das zahlenmäßig erfasste Risiko ein vorgegebenes Grenzrisiko nicht überschreitet. Dabei wird vorausgesetzt, dass ein irgendwie gearteter sozialer Prozess das akzeptable Risiko offenbart hat. Die Entscheidungen im Rahmen der technischen Entwicklung werden dann rein sachlich auf der Grundlage der Risikoanalyse getroffen. Vorherrschend ist die probabilistische Denkweise. Die Erfahrungen der Vergangenheit stecken in den Statistiken zu den Fehlerhäufigkeiten und den Schadensauswirkungen sowie in den wissenschaftlich geprüften Modellen zur Ermittlung von Kennzahlen. Der soziale Prozess - die Werte- und Zieldiskussion zur Festlegung des akzeptablen Risikos - ist der eigentlichen Sicherheitstechnik vorgelagert. Sicherheitstechnik wird also auf den technisch-wissenschaftlichen Bereich reduziert. Sicherheit ist ein Analyseergebnis; sie ist objektiv.

In Kürze: Die qualitative Methode beruht auf der ausgiebigen Nutzung des Wissens der Beteiligten durch strukturierte soziale Prozesse. Die quantitative Methode setzt auf die Objektivierung des Wissens mittels Statistiken und Theorien.

Der Trend hin zur Quantifizierung von Sicherheit ist verbunden mit der Einführung neuer Techniken mit hohem Gefährdungspotential wie Kerntechnik und Magnetschwebebahn. Inzwischen gibt es gut akzeptierte Richtlinien und Normen zu diesem Thema: DIN VDE 31 000/2 und VDI/VDE 3542/2 sind Beispiele dafür.

Aber die Quantifizierung hat ihre Grenzen. Hier einige der Argumente, die zur Vorsicht mahnen.

- Die quantitative Analyse fußt auf der qualitativen. Das gilt für die Fehlerbaumanalyse (Kapitel 7) genauso wie für die Ereignisbaumanalyse (Kapitel 8). Stets kommt zuerst eine qualitative Analyse der Zusammenhänge zwischen Fehlerursachen und Wirkungen; erst dann erfolgt eine quantitative Bewertung.
- Die Frage nach dem vertretbaren Risiko, also die Frage „Wie sicher ist sicher genug?“, lässt sich nicht allgemein beantworten. Die Gesellschaft ist kaum in der Lage, sich auf ein Grenzrisiko zu einigen und der Technik ein solches vorzugeben.
- Fehleinschätzungen werden durch Quantifizierung verdeckt. Die ermittelten Zahlen täuschen Objektivität oft nur vor.
- Wichtige Faktoren lassen sich nicht einfach mit Zahlen belegen. Das gilt besonders für Entwurfs- und Programmierfehler.
- Der Reiz der Quantifizierung kann die Aufmerksamkeit von einfacheren und wirkungsvolleren Techniken ablenken.
- Manchmal ist die Quantifizierung nur eine Ausflucht aus der Mühsal, ein System verstehen zu müssen.

Die qualitative Sicherheitstechnik lässt sich ebenfalls nicht in Reinkultur verwirklichen. Ganz ohne Quantifizierung geht es auch hier nicht. Irgendwann muss beispielsweise einmal gesagt werden, was als *geringe* und was als *hohe* Eintrittswahrscheinlichkeit für einen Schadensfall anzusehen ist.

In der Praxis wird man immer Mischformen aus qualitativer und quantitativer Sicherheitstechnik vorfinden. In diesem Kapitel geht es erst einmal um die qualitative Sicherheitstechnik.

6.1 Anforderungsklassen

Zentraler Begriff der qualitativen Sicherheitstechnik ist die *Anforderungsklasse*. Sicherheitstechnisch relevante Einrichtungen werden im Rahmen ihrer Spezifizierung in eine der Anforderungsklassen eingeteilt. Mit der Anforderungsklasse sind sicherheitstechnische Maßnahmen verbunden. Welche Maßnahmen das sind, hängt von dem jeweiligen Anwendungsgebiet ab.

Hier wollen wir uns die Anforderungen an Schutzeinrichtungen im Rahmen der Prozessleittechnik - kurz: PLT-Schutzeinrichtungen - etwas genauer ansehen (VDI/VDE-Richtlinie 2180; Weidlich, 1997; VDE/VDI-AG, 1987). Diese spezielle Anwendung soll nur soweit studiert werden, wie es dem Verständnis der Vorgehensweise dient, und es werden solche Dinge angesprochen, die man ganz ähnlich auch auf anderen Anwendungsgebieten findet. Über die speziell auf die Qualitätssicherung von Software zugeschnittenen Richtlinien berichten Hohler und Villinger (1998).

Zweck der PLT-Schutzeinrichtungen ist die Verhinderung von Störfällen in verfahrenstechnischen Anlagen. Sie werden eingesetzt, wenn ansonsten mit Fehlerzuständen der verfahrenstechnischen Anlage gerechnet werden muss, die unmittelbar zu Personenschäden, Umweltschäden oder Sachschäden führen können.

Die sicherheitstechnischen Festlegungen werden durch eine interdisziplinäre Arbeitsgruppe getroffen und in einem von allen Teilnehmern unterschriebenen Dokument festgehalten.

Risiko-Einflussgrößen	
Schadensausmaß	
S1	leichte Verletzung einer Person oder kleinere schädliche Umwelteinflüsse, die z.B. nicht unter die Störfallverordnung fallen
S2	schwere, irreversible Verletzung einer oder mehrerer Personen oder Tod einer Person oder vorübergehende größere schädliche Umwelteinflüsse
S3	Tod mehrerer Personen oder lang andauernde größere schädliche Umwelteinflüsse
S4	katastrophale Auswirkung, sehr viele Tote
Aufenthaltsdauer	
A1	selten bis öfter
A2	häufig bis dauernd
Gefahrenabwendung	
G1	möglich unter bestimmten Bedingungen
G2	kaum möglich
Eintrittswahrscheinlichkeit	
W1	sehr gering
W2	gering
W3	relativ hoch

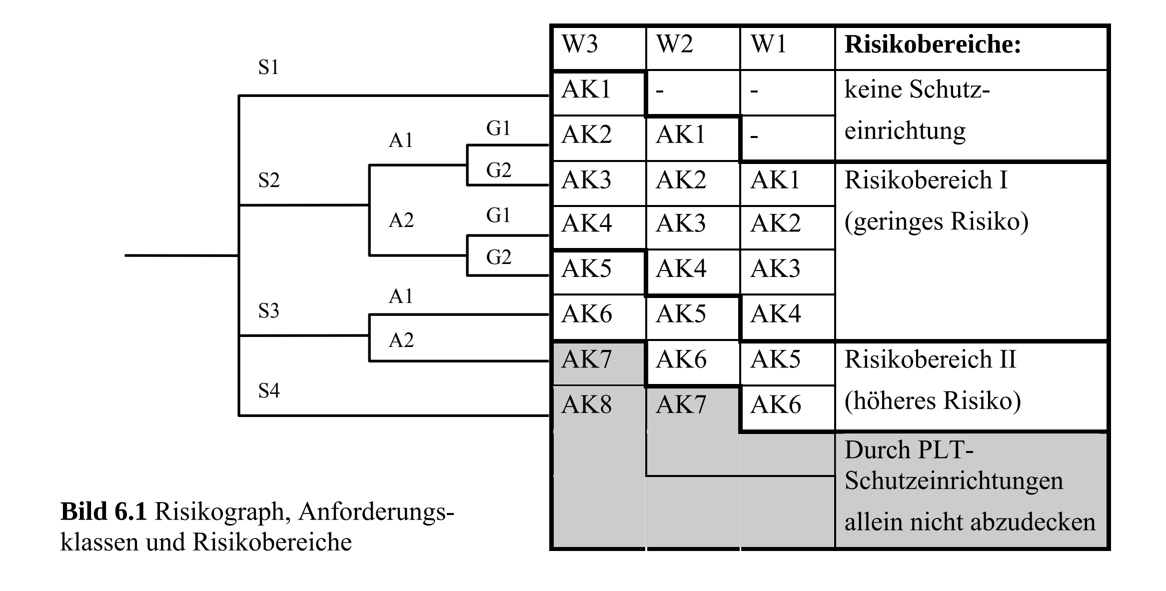


Bild 6.1 Risikograph, Anforderungsklassen und Risikobereiche

Mitwirkende sind

- der Betreiber,
- der Anlagenbauer,
- die Fachabteilungen für Entwicklung und Konstruktion, sowie
- die für die Qualitätssicherung zuständige Stelle.

Zu den Aufgaben der Qualitätssicherung gehört, dass die relevanten technischen Regeln, Richtlinien, Vorschriften, Normen und Gesetze beachtet bzw. eingehalten werden. Unabhängige Gutachter (z.B. von einem Technischen Überwachungs-Verein) werden hier beratend tätig.

Das Sicherheitsgespräch durchläuft drei Schritte:

1. Qualitative Abschätzung des abzudeckenden Risikos.
2. Festlegung der Anforderungen, Bestimmung der Anforderungsklasse.
3. Zuordnung technischer und organisatorischer Maßnahmen.

Das abzudeckende Risiko wird mit Hilfe des Risikographen ermittelt (Bild 6.1 und Kasten „Risiko-Einflussgrößen"). Dazu werden Schadensausmaß, Aufenthaltsdauer, die Möglichkeiten der Gefahrenabwehr und die Eintrittswahrscheinlichkeit des Schadensfalls qualitativ abgeschätzt. Der Risikograph ordnet diesem abzudeckenden Risiko eine der Anforderungsklassen AK1 bis AK8 zu. Die Anforderungsklassen sind in Risikobereiche eingeteilt. Für diese Risikobereiche existieren generelle Anforderungen an die technische Einrichtung – sozusagen eine Rahmenspezifikation.

Der Risikobereich I (geringes Risiko) umfasst die Anforderungsklassen AK1 bis AK4. Einrichtungen dieser Klassen müssen folgende Anforderung erfüllen: Ein passiver Fehler muss innerhalb einer Zeitspanne erkannt und beseitigt werden, in der nicht gleichzeitig mit der Störung des bestimmungsgemäßen Betriebs vernünftigerweise gerechnet werden muss.

Der Risikobereich II (höheres Risiko) umfasst die Anforderungsklassen AK5 und AK6 und stellt die folgenden Anforderungen: Ein passiver Fehler darf die Fähigkeit der PLT-Schutzeinrichtung zur Ausführung der Schutzfunktion nicht beeinträchtigen. Er muss unabhängig vom Prozessverhalten in einer Zeitspanne erkannt und beseitigt werden, in der nicht gleichzeitig mit dem Auftreten eines zweiten unabhängigen Fehlers vernünftigerweise gerechnet werden muss.

Die Anforderungsklassen AK7 und AK8 sind nicht mehr allein mit Schutzmaßnahmen im Rahmen der Prozessleittechnik abzudecken. Hier sind spezielle Schutzmaßnahmen vorgeschrieben, die besonders strengen Anforderungen genügen müssen.

Den Risikobereichen sind technische und organisatorische Maßnahmen zugeordnet. Speziell für den Bereich des höheren Risikos sind das

- Redundanz der Ein- und Ausgabe, möglichst in Form physikalisch unterschiedlicher Prozessvariablen und mit diversitären Geräten;
- selbstüberwachende Logik oder 1-aus-2-Schaltung;
- Unabhängigkeit der PLT-Schutzeinrichtung von den PLT-Betriebseinrichtungen insoweit, als bei Ausfall von PLT-Betriebseinrichtungen die Funktion der PLT-Schutzeinrichtung erhalten bleibt.

6.2 Sicherheitsorientierte Entwicklung – Design for Safety

Die sicherheitsorientierte Entwicklung will Gefahren identifizieren, diese möglichst eliminieren oder zumindest minimieren und das, was an Gefahren übrig bleibt, beherrschen.

Die Gefahr, dass ein Gerät durch Überhitzung ausfällt, weil der Ventilator defekt ist, lässt sich möglicherweise dadurch *eliminieren*, dass man Kühlung durch Konvektion nutzt.

Gefahren lauern in schlecht durchschaubaren Systemen aus vielen miteinander eng verkoppelten Komponenten. Vor allem das Überladen einzelner Komponenten eines Systems mit verschiedenen Funktionen kann dramatische Folgen haben. Nancy Leveson (1995) bringt das Beispiel einer Chemie-Anlage, in der ein verzweigtes Röhrensystem für den Transport verschiedener Flüssigkeiten genutzt wird. Die Steuerung der Ventile der komplexen Anlage erfolgt durch einen Computer. Durch Handeingriff kam es in dieser Anlage einmal zu einer Störung, bei der Methanol in die Umwelt gelangte. Mit einem *einfacheren* Design, das für die verschiedenen Stoffe separate Leitungen vorsieht, kann so etwas verhindert werden. Und eine solche Alternative kann, über die gesamte Lebenszeit der Anlage gesehen, sogar billiger sein. Der einfachere Entwurf und die *Entflechtung* der Funktionen helfen, Gefahren zu vermeiden.

Zur Reduzierung der Komplexität der kritischen Teile werden die sicherheitsrelevanten Teile auf das Allernötigste beschränkt und zu eigenen Modulen zusammengefasst. Die kritischen werden von den unkritischen Funktionen möglichst *entkoppelt* und durch *Barrieren* getrennt, so dass die unkritischen Module das System nicht in einen gefährlichen Zustand bringen können.

Letztlich führt dieser Entwurfsprozess zu einer Identifizierung von Einrichtungen mit Sicherheitsverantwortung. Dazu gehören die PLT-Schutzeinrichtungen des vorhergehenden Abschnitts. Zu den besonderen Anforderungen an diese Einrichtungen gehört, dass ein einfacher Komponentenausfall nicht zum gefährlichen, das heißt sicherheitsrelevanten Ausfall der Einrichtung führen darf, und dass ein solcher Komponentenausfall in hinreichend kurzer Zeit erkannt und beseitigt wird.

Fehlertolerante sicherheitsgerichtete Schaltungen

Das Thema der sicherheitsorientierten Schaltungstechnik wird hier auf dem relativ hohen Abstraktionsniveau der logischen (booleschen) Verknüpfungen angesprochen. Die Schaltungsbeispiele sollen deutlich machen, mit welchen grundlegenden Problemen man es auf diesem Gebiet zu tun hat. In der Praxis erfordert die Sicherheitstechnik ein tieferes Eindringen in die Technologie und die physikalischen und chemischen Effekte (Trompeta, Wettingfeld, 1996).

Sehen wir uns eine Logikschaltung an, die der Steuerung oder Überwachung eines Prozesses dient. Ihre Spezifikation verlangt, dass einfache Komponentenfehler nicht zu einer unbemerkten Fehlfunktion führen, sondern dass sie gemeldet werden, um das Gesamtsystem in einen sicheren Zustand überführen zu können (Fail-Safe-Verhalten). Einfache Fehler in nicht funktionsbeteiligten Teilen der Schaltung müssen ebenfalls in hinreichend kurzer Zeit erkannt und gemeldet werden, und zwar unabhängig vom Prozessverhalten. Von der Schaltung wird nicht gefordert, dass sie ihre Funktion auch im Fehlerfall aufrechterhält.

Die Anforderungen legen nahe, die Logikschaltung als 1-aus-2-System auszulegen. Bei Logikschaltungen ist es üblich, die beiden Subsysteme des 1-aus-2-Systems in komplementärer Logik auszuführen. Der Ausgangsgrößen sind dann nicht auf Gleichheit, sondern auf Antivalenz der Signale zu überwachen.

Die Grundstruktur dieser fehlertoleranten Logikschaltung zeigt Bild 6.2. Sie besteht aus einem Modul, das die eigentliche Funktion realisiert (Funktionslogik). Dem wird ein „zweiter Kanal"

in komplementärer Logik hinzugefügt. Zur Überprüfung der Funktion werden aus der Funktionslogik die Signale e_1, e_2, e_3, ..., e_n und aus der komplementären Logik die dazu komplementären (logisch negierten) Signale e'_1, e'_2, e'_3, ... , e'_n herausgeführt. Diese Signale werden einer Antivalenzüberwachung zugeführt. Letztere spielt die Rolle des Vergleichers im 1-aus-2-System.

Aufgabe der Antivalenzüberwachung ist es, alle Signalpaare (e_i, e'_i) auf Antivalenz zu überwachen. Eine Fehlermeldung soll immer dann erfolgen, wenn für wenigstens eins der Signalpaare die Antivalenzbedingung $e'_i = \neg e_i$ nicht erfüllt ist ($\neg$ bezeichnet die Negation). Im fehlerfreien Fall gibt die Überwachungsschaltung ein antivalentes Signalpaar ab: $a' = \neg a$. Eine Fehlermel-

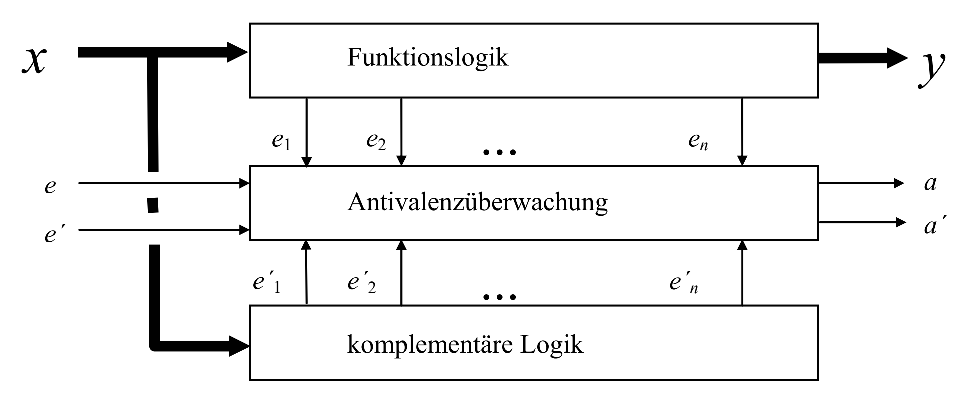


Bild 6.2 Zweikanaliger Aufbau einer logischen Funktion

dung liegt vor, wenn das Signalpaar am Ausgang äquivalent ist: $a' = a$. Wir nennen eine solche Schaltung, die fehlerhafte Eingänge erkennt und meldet, *fehlererkennend*.

Was aber passiert, wenn die Antivalenzüberwachung selbst defekt ist? Gemäß Sicherheitsvereinbarung muss das Überwachungsmodul auch einfache Fehler in der eigenen Einheit in hinreichend kurzer Zeit melden. Dabei darf dann allerdings die korrekte Funktion der übrigen Teile vorausgesetzt werden.

Bild 6.3 zeigt den Aufbau einer fehlererkennenden Überwachungseinheit für zwei Eingangsgrößen *A* und *B* (Sedmak, Liebergot, 1978). Ist die Überwachungseinheit selbst fehlerfrei, dann liefert sie immer ein antivalentes Ausgangssignalpaar, solange die Eingangssignalpaare antivalent - also fehlerfrei - sind. Falls wenigstens eins der Eingangssignalpaare äquivalent ist, ergibt sich auch ein äquivalentes Ausgangssignalpaar, also eine Fehlermeldung. Diese Einheiten lassen sich zu größeren Einheiten für die Überwachung mehrerer Signalpaare zusammenschalten. Die so entstehende Antivalenzüberwachung ist insgesamt fehlererkennend.

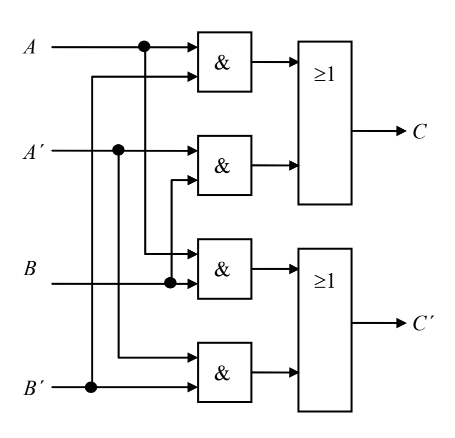


Bild 6.3 Fehlersichere Überwachungseinheit

Die Überwachungseinheit ist aber auch *selbstüberwachend* im folgenden Sinne: Jeder einfache Fehler eines Logikbausteins, bei dem die Ausgangsvariable fest auf 0 (stuck-at-0) oder fest auf 1 (stuck-at-1) ist, wird zumindest bei einer bestimmten Kombination von fehlerfreien Eingangssignalen erkannt. Nehmen wir einmal an, dass das oberste der Und-Gatter (&) fest auf 0

liegt. Die Eingangskombination A=1 und B=0 (zusammen mit den korrekten Komplementen A'=0 und B'=1) liefert in diesem Fall das fehleroffenbarende Ausgangssignalpaar C=0 und C'=0.

Die Anforderungen bezüglich der Fehlererkennung verlangen, dass die Eingangsvariablen in ausreichend kurzer Zeit alle fehleroffenbarenden Eingangskombinationen durchlaufen. Die Eingangskombinationen hängen aber von den Prozessvariablen ab.

Für eine von den Prozessvariablen unabhängige Fehlererkennung könnte man zu einer Dynamisierung der Eingangsvariablen der Überwachungsschaltung übergehen. Dazu wird jedes Eingangssignalpaar mittels zweier Exklusiv-Oder-Gattern mit einem dritten Signal I verknüpft, das für I=1 eine Invertierung des Signalpaares besorgt und es für I=0 unverändert lässt, Bild 6.4.

Die Dynamisierungsschaltung ist für jeden festen Wert I fehlererkennend und sie ist selbstüberwachend, und zwar unabhängig von Signalwechseln der Größe A. Mit mehreren solcher Invertierungssignale lässt sich erreichen, dass die Eingänge der Antivalenzüberwachung alle möglichen fehleroffenbarenden Eingangskombinationen durchlaufen.

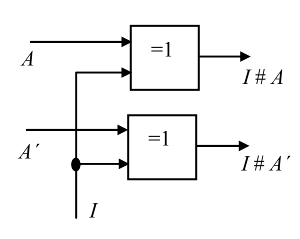


Bild 6.4 Dynamisierung

6.3 Ausfalleffektanalyse

Die Ausfalleffektanalyse bildet die Basis des Nachweises, dass eine technische Einrichtung die an sie gestellten Zuverlässigkeits- und Sicherheitsanforderungen erfüllt. Auf ihr bauen beispielsweise die Fehlerbaumanalyse und die Ereignisbaumanalyse auf. In einfachen Fällen reicht die Ausfalleffektanalyse für den Nachweis aus.

Die *Ausfalleffektanalyse* (FMEA, Failure Modes and Effects Analysis) wird im Team durchgeführt, dem wenigstens der Entwicklungsingenieur und ein Zuverlässigkeitsingenieur angehören. Sie ist eine Bottom-Up-Methode: Für jede Komponente des zu analysierenden Systems und für jede zu berücksichtigende Fehlerart der Komponente wird die Konsequenz bezüglich der spezifizierten Funktion bzw. bezüglich der Sicherheit tabellarisch erfasst (Kuhlmann, 1981, Birolini, 1991).

Die Tabelle enthält also mindestens die Spalten für (1) die laufende Nummer des Eintrags, (2) die eindeutige Benennung der Komponente, (3) die Bezeichnung des Komponententyps, (4) die Fehlerart der Komponente und (5) die Fehlerauswirkungen.

Bei der Beurteilung der Fehlerauswirkungen wird eine Klassifizierung nach der Schwere der Auswirkungen vorgenommen. Liegt eine Sicherheitsspezifikation vor, dann ist insbesondere festzuhalten, ob es sich um einen sicherheitsrelevanten Fehler handelt oder nicht.

Zusätzliche Spalten können vorgesehen werden für die Erfassung

- der weiterreichenden Auswirkungen, beispielsweise auf die zu erfüllende Mission,
- der Maßnahmen zur Minderung der Fehlerfolgen und
- der Schutzaktionen.

Oft wird die Ausfalleffektanalyse mit einer einfachen Quantifizierung verbunden. Dann wird für jede zu berücksichtigende Fehlerart des Systems eine Spalte angelegt, in der die Ausfallraten der Komponentenausfälle einzutragen sind, die genau zu dieser Fehlerart beitragen.

Die Anregung zu dem folgenden einfachen Beispiel habe ich aus dem Buch von Nancy Leveson (1995). Das System besteht aus der Parallelschaltung zweier Verstärker A und B, Bild 6.5. Die Verstärker mögen eine Ausfallrate von je 10^{-6}/h besitzen. Ungefährlich ist der Ausfall eines Verstärkers, bei dem der Ausgang des Verstärkers zum Leerlauf wird. Das möge in 90 % der Fälle geschehen. Ein Kurzschluss kommt in 5 % der Fälle vor. Ungewiss ist, was in den restlichen 5 % der Fälle passiert. Der ungünstigste Fall wird angenommen. Tabelle 6.1 zeigt die Ausfalleffektanalyse zusammen mit den Ausfallraten.

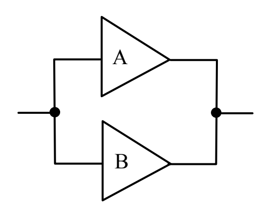


Bild 6.5 Parallelgeschaltete Verstärker

Die sicherheitsbezogene Ausfallrate des Gesamtsystems erhält man durch Addition der Ausfallraten in der letzten Spalte. Sie ist gleich $2\cdot10^{-7}$/h. Die Ausfallratenaddition liefert auch dann noch ein sinnvolles und gut interpretierbares Ergebnis, wenn man – wie hier geschehen – die Komponentenausfallraten auf die sicherheitsrelevanten Anteile reduziert (Grams/Angermann, 1981).

Tabelle 6.1 Ausfalleffektanalyse

Nr.	*Komponente*	*Typ*	*Fehlerart*	*Auswirkung* 1: ungefährlich 2: gefährlich	*Ausfallrate* (sicherheits-bezogen)
1	A	XXX	Leerlauf	1	
2			Kurzschluss	2	10^{-6}/h·5%
3			anderes	2	10^{-6}/h·5%
4	B	XXX	Leerlauf	1	
5			Kurzschluss	2	10^{-6}/h·5%
6			anderes	2	10^{-6}/h·5%

6.4 Fehlerbetrachtung

Eine Schaltung wird nicht allein dadurch schon wesentlich sicherer, dass sie fehlertolerant ist. Wenn der Kehrwert der Systemausfallrate etwa so groß wie die Betriebsdauer ist, muss man damit rechnen, dass es zu zwei Fehlern innerhalb der Betriebsdauer kommt. Eine grobe Abschätzung mit dem Modell des Poisson-Stromes zeigt, dass die Wahrscheinlichkeit für diesen sicherheitskritischen Fall etwa gleich 45 % ist. Dabei wird vorausgesetzt, dass sich die Ausfallrate nach dem ersten Ausfall praktisch nicht ändert, und dass es mit unveränderter Übergangsrate auch zum zweiten Ausfall kommt.

Die detaillierte quantitative Analyse (Kapitel 10) wird zeigen, dass sich Fehlertoleranz erst auszahlt, wenn entweder

- die *Missionsdauer* weit unterhalb der mittleren Lebensdauer des Systems liegt, oder wenn
- *Fehler selbsttätig erkannt* und gemeldet werden, so dass das System innerhalb einer hinreichend kurzen Zeitspanne repariert werden kann, oder

- wenn das System in hinreichend kurzen Abständen einer *Wartung* unterzogen wird, bei der eventuell vorhandene - sich nicht selbst offenbarende - Fehler entdeckt und beseitigt werden können.

Die *Instandhaltungsstrategie* für eine Steuerungseinrichtung mit vereinbarter gesicherter Funktion wird über eine *Fehlerbetrachtung* festgelegt. Für jeden in Betracht zu ziehenden (sicherheitsrelevanten) Komponentenfehler läuft diese Fehlerbetrachtung im Kern wie im Ablaufplan des Bildes 6.6 ab. Es handelt sich dabei um eine gekürzte Fassung der Fehlerbetrachtung aus VDI/VDE 3541/4. Die Fehlerbetrachtung zeigt auch, ob das System prinzipiell für die Sicherheitsaufgabe geeignet ist.

6.5 Technik und Recht

Die Abwendung von Gefahren für die Öffentlichkeit ist eine Aufgabe des Staates und des Rechtssystems. Demzufolge geht es in der Sicherheitstechnik auch um Rechtsfragen. Vieles leitet sich bereits aus der Schadensersatzpflicht ab, wie sie im Absatz 1 des § 823 im Bürgerlichen Gesetzbuches dargelegt wird: „Wer vorsätzlich oder fahrlässig das Leben, den Körper, die Gesundheit, die Freiheit, das Eigentum oder ein sonstiges Recht eines anderen widerrechtlich verletzt, ist dem anderen zum Ersatze des daraus entstehenden Schadens verpflichtet."

Zuallererst ist für die Gefahrenabwehr private Vorsorge zu treffen. Diesem Zweck dienen beispielsweise die Technischen Überwachungs-Vereine, die ab 1866 als Zusammenschluss der Dampfkesselbetreiber gegründet worden sind. Sie machten die behördliche Überwachung entbehrlich.

Aber darüber hinaus gibt es abgestufte staatliche Eingriffsmöglichkeiten, die in Gesetzen festgelegt sind.

Das Gerätesicherheitsgesetz legt im § 3, Abs. 1, u. a. fest: „Technische Arbeitsmittel ... dürfen nur in den Verkehr gebracht werden, wenn sie nach den allgemein anerkannten Regeln der Technik sowie den Arbeitsschutz- und Unfallverhütungsvorschriften so beschaffen sind, daß Benutzer oder Dritte bei ihrer bestimmungsgemäßen Verwendung gegen Gefahren aller Art für Leben oder Gesundheit soweit geschützt sind, wie es die Art der bestimmungsgemäßen Verwendung gestattet."

Werden, wie hier geschehen, die *allgemein anerkannten Regeln der Technik* zu Grunde gelegt, können sich Behörden und Gerichte darauf beschränken, die herrschende Auffassung unter den technischen Praktikern zu ermitteln. Sie hinkt gewöhnlich hinter der technischen Entwicklung her. Das Gerätesicherheitsgesetz beschränkt das behördliche Einschreiten auf den Fall, dass sich die Unsicherheit eines Gerätes offenbart hat.

Strengere Regelungen enthält beispielsweise das Bundes-Immissionsschutz-Gesetz. Die davon betroffenen Anlagen sind genehmigungspflichtig, und außerdem wird bei der Beurteilung der *Stand der Technik* zu Grunde gelegt. Letzterer bezieht sich auf die Front der Entwicklung und kann unter Umständen strittig sein.

Laut Atomgesetz ist die Genehmigung einer entsprechenden Anlage nur zulässig, wenn die erforderliche Vorsorge gegen Schäden getroffen ist. Ein Anspruch auf atomrechtliche Genehmigung besteht nicht. Die Beurteilung der Anlagen legt den *Stand von Wissenschaft und Technik* zu Grunde. Hier wird die Vorsorge also nicht durch das technisch Machbare begrenzt. Eventuell wird eine Genehmigung verweigert.

Technische Regeln und Normen haben zunächst keine rechtliche, sondern "tatsächliche" Qualität: sie werden aus der Praxis heraus entwickelt (Kuhlmann, 1981, S. 381 ff.). Folgendermaßen können sie rechtliche Bedeutung erlangen:

1. Technische Regeln können durch Gesetze oder Verordnungen "inkorporiert" - das heißt wörtlich übernommen - werden.
2. Im Gesetzestext kann ein Verweis auf Normen stehen.
3. Generalklauseln können auf die „Regeln“ oder den „Stand ...“ verweisen.

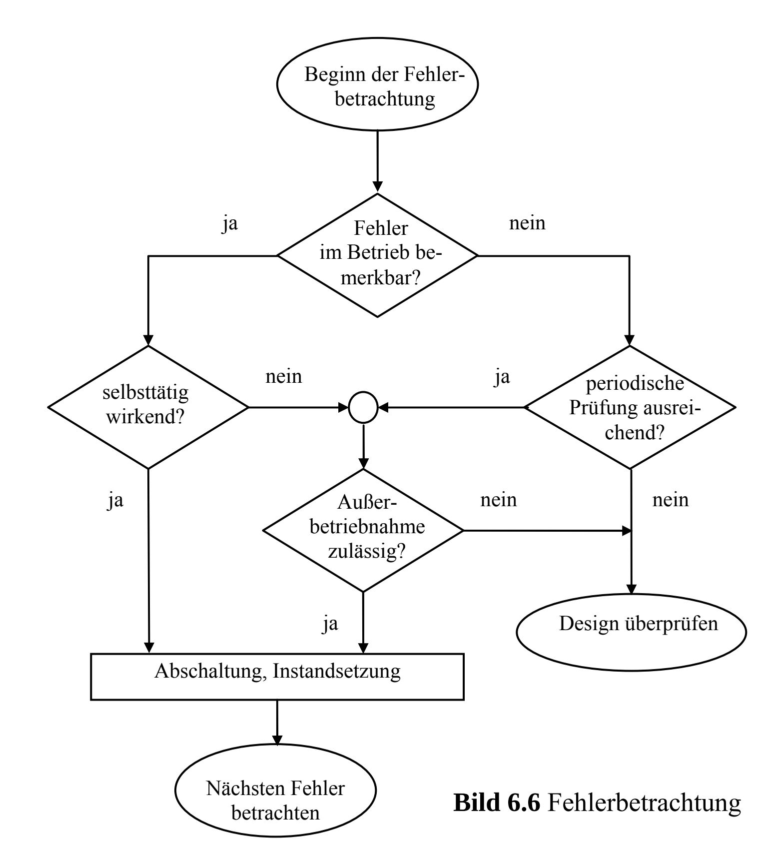


Bild 6.6 Fehlerbetrachtung

7 Fehlerbaumanalyse

Die *Fehlerbaumanalyse* (FTA, Fault Tree Analysis) ist eine Methode zur systematischen Erfassung aller möglichen Ursachen eines bestimmten Top-Ereignisses (Andrews/Moss, 1993; Shooman, 1990; Schneeweiss, 1989). Angewendet wurde die Methode im großen Stil in den Reaktorsicherheitsstudien (BMFT, 1980; Lewis, 1980).

Ausgehend von einem Top-Ereignis werden alle möglichen unmittelbaren Ursachen ermittelt. Für jedes der ursächlichen Ereignisse wird dieser Schritt wiederholt, bis hin zu den Basisereignissen. Die Analyse ist rückwärts gewandt. Insofern ist sie der Ereignisbaumanalyse entgegengesetzt (Kapitel 8), die von der Ursache ausgehend nach den möglichen Auswirkungen fragt.

Die Fehlerbaumanalyse beruht auf der binären Beschreibung von Komponentenzuständen: intakt/defekt, ein/aus, offen/geschlossen ... Das erlaubt die Nutzung der booleschen Algebra und der Logik. Deshalb wird im nächsten Abschnitt erst einmal eine der Zuverlässigkeitstechnik angemessene Einführung in die wesentlichen Definitionen und Rechenregeln der booleschen Algebra gebracht.

Die grafische Darstellung von Fehlerbäumen ist standardisiert. Sie ist angelehnt an die Diagramme für boolesche Schaltnetze (amerikanische Norm). Speziell für Software wurde die Software-Fehlerbaum-Analyse (Software Fault Tree Analysis, SFTA) entwickelt (Leveson, 1995; Lyu, 1996).

7.1 Boolesche Algebra

Binäre Funktionen binärer Variablen heißen *Schaltfunktionen*. Sei f eine Schaltfunktion mit n binären Variablen:

$$f\colon B^n \to B$$

mit $B = \{0, 1\}$. Die Werte 0 und 1 werden normalerweise mit den Wahrheitswerten falsch und wahr identifiziert. Anstelle von Schaltfunktionen spricht man zuweilen auch von *booleschen Funktionen*. Boolesche Operatoren lassen sich mit Hilfe der arithmetischen Operatoren definieren. Hier die wichtigsten dieser Definitionen, nämlich die für Konjunktion, Disjunktion und Negation:

$$\mathrm{AND}(a, b) = a \wedge b = ab \qquad \text{(Konjunktion)}$$

$$\mathrm{OR}(a, b) = a \vee b = a + b - ab \qquad \text{(Disjunktion)}$$

$$\mathrm{NOT}(a) = \neg a = 1 - a \qquad \text{(Negation)}$$

Statt $a \wedge b$ wird zuweilen auch a & b oder a AND b geschrieben. Statt $a \vee b$ steht manchmal auch a OR b. Die Verwendung des andernorts gebräuchlichen +-Zeichens verbietet sich hier, da wir Arithmetik-Operatoren und Logikoperatoren freizügig mischen wollen. Die Vorrangregeln und die Regeln der Klammernrechnung folgen den üblichen Regeln der Arithmetik, wobei die Negation dem negativen Vorzeichen gleichrangig ist, die Konjunktion der Multiplikation und die Disjunktion der Addition.

Für das Rechnen mit booleschen Ausdrücken (das sind mit booleschen Operatoren zusammengesetzte Ausdrücke) gelten die Transformationsregeln der *booleschen Algebra*. Sie lassen sich leicht mit den Regeln der Zahlenrechnung beweisen, wenn man die obigen Definitionsglei-

chungen verwendet und berücksichtigt, dass für die hier zu Grunde gelegten Werte 0 und 1 stets $aa = a$ gilt. Hier die wichtigsten der Regeln:

1. Kommutativgesetze

$a \wedge b = b \wedge a$

$a \vee b = b \vee a$

2. Assoziativgesetze

$(a \wedge b) \wedge c = a \wedge (b \wedge c)$

$(a \vee b) \vee c = a \vee (b \vee c)$

3. Distributivgesetze

$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

4. Absorptionsgesetze

$a \wedge (a \vee b) = a$

$a \vee (a \wedge b) = a$

5. Existenz neutraler Elemente

$0 \vee a = a$

$1 \wedge a = a$

6. Existenz komplementärer Elemente

$a \vee \neg a = 1$

$a \wedge \neg a = 0$

7. *De Morgansche Gesetze*

$\neg(a \wedge b) = \neg a \vee \neg b$

$\neg(a \vee b) = \neg a \wedge \neg b$

Nur die De Morganschen Gesetze sollen hier einmal hergeleitet werden. Wegen

$\neg a \vee \neg b = (1-a)+(1-b) - (1-a)(1-b) = 1- ab = \neg(a \wedge b)$

gilt das erste der De Morganschen Gesetze. Das zweite folgt aus

$\neg a \wedge \neg b = (1-a)(1-b) = 1-(a+b-ab) = \neg(a \vee b)$.

Jede boolesche Funktion kann als *disjunktive Normalform* (DNF) dargestellt werden. Das ist eine Disjunktion *einfacher Terme.* Ein einfacher Term ist eine Konjunktion ohne Klammern, der nur Variablen oder negierte Variablen enthält, und in dem jede Variable höchstens einmal vorkommt, so wie hier: $x \wedge \neg y \wedge z$, oder kurz $x \neg y z$.

Unter einer *Vollkonjunktion* wird ein einfacher Term verstanden, in der jede unabhängige Variable einer Funktion genau einmal vorkommt. Eine solche Vollkonjunktion nimmt für genau eine Wertebelegung den Wert eins an und sie ist gleich null für alle anderen Wertebelegungen. Eine DNF, die nur Vollkonjunktionen enthält, heißt kanonische disjunktive Normalform (CDNF).

Aus der Wertetabelle einer Funktion lässt sich direkt ihre CDNF entwickeln: Für jede Zeile, in der das Funktionsergebnis gleich eins ist, wird eine Vollkonjunktion gebildet, die genau für die Wertebelegung dieser Zeile den Wert eins annimmt. Die disjunktive Verknüpfung dieser Vollkonjunktionen liefert das gewünschte Ergebnis.

Beispiele: Tabelle 7.1 enthält die Wertetabellen der Funktionen IMPL(x, y) und EQUAL(x, y) und EXOR(x, y). Für die Implikation IMP(x, y) schreibt man auch $x \leq y$, anstelle von EQUIV(x, y) auch $x = y$ und anstelle von EXOR(x, y) auch $x \# y$.

Tabelle 7.1 Wertetabellen

x	y	IMP(x, y)	EQUIV (x, y)	EXOR(x, y)
0	0	1	1	0
0	1	1	0	1
1	0	0	0	1
1	1	1	1	0

Für die Funktion IMP ergibt sich die CDNF folgendermaßen: Die Vollkonjunktion $\neg x \wedge \neg y$ ergibt nur für die Wertebelegung der ersten Zeile eine 1, die Vollkonjunktion $\neg x \wedge y$ nur in der zweiten und $x \wedge y$ nur in der vierten Zeile. Die disjunktive Verknüpfung dieser Vollkonjunktionen liefert IMP(x, y) = $\neg x \wedge \neg y \vee \neg x \wedge y \vee x \wedge y$. Mit den Transformationsregeln erhält man daraus eine minimierte DNF:

$$\text{IMP}(x, y) = (x \leq y) = \neg x \vee y \qquad \text{(Implikation)}$$

Entsprechendes ergibt sich für die Gleichheit (Äquivalenz) und die Ungleichheit (Antivalenz, Exklusives Oder):

$$\text{EQUIV}(x, y) = (x = y) = x \wedge y \vee \neg x \wedge \neg y \qquad \text{(Äquivalenz)}$$

$$\text{EXOR}(x, y) = (x \# y) = \neg x \wedge y \vee x \wedge \neg y \qquad \text{(Antivalenz)}$$

Da sich offenbar grundsätzlich jede Schaltfunktion auf diese Weise durch die Operationen AND, OR und NOT darstellen lässt, wird diese Verknüpfungsmenge als *funktional vollständig* bezeichnet.

Schließlich führen wir noch zwei Operatoren ein, von denen jeder für sich funktional vollständig ist: NAND(x, y) = $\neg(x \wedge y)$ und NOR(x, y) = $\neg(x \vee y)$. Die funktionale Vollständigkeit des NAND ergibt sich beispielsweise aus folgenden Darstellungen für AND, OR und NOT:

$$\text{NOT}(x) = \text{NAND}(x, x)$$

$$\text{AND}(x, y) = \text{NAND}(\text{NAND}(x, y), \text{NAND}(x, y))$$

$$\text{OR}(x, y) = \text{NAND}(\text{NAND}(x, x), \text{NAND}(y, y))$$

Offensichtlich kann man allein mit NAND-Verknüpfungen jede Schaltfunktion darstellen. Den Beweis für die Formeln erhält man leicht durch Einsetzen der Definitionsgleichungen und Vereinfachungen nach den Regeln der booleschen Algebra.

7.2 Der Fehlerbaum

Bild 7.1 zeigt ein P&I-Diagramm eines Kraftwerks (P&I steht für *Pipes and Instruments*). In dieser Anlage sind zwei Pumpen parallel angeordnet. Ihre Aufgabe ist es, den Primärkreislauf in Gang zu halten. Dieser transportiert die Wärmeenergie von zwei gasbefeuerten Heizkesseln (Boiler) zu den Wärmetauschern (Exchanger). Diese versorgen den Sekundärkreislauf mit Energie, die dazu dient, die Turbinen für die Elektrizitätserzeugung anzutreiben.

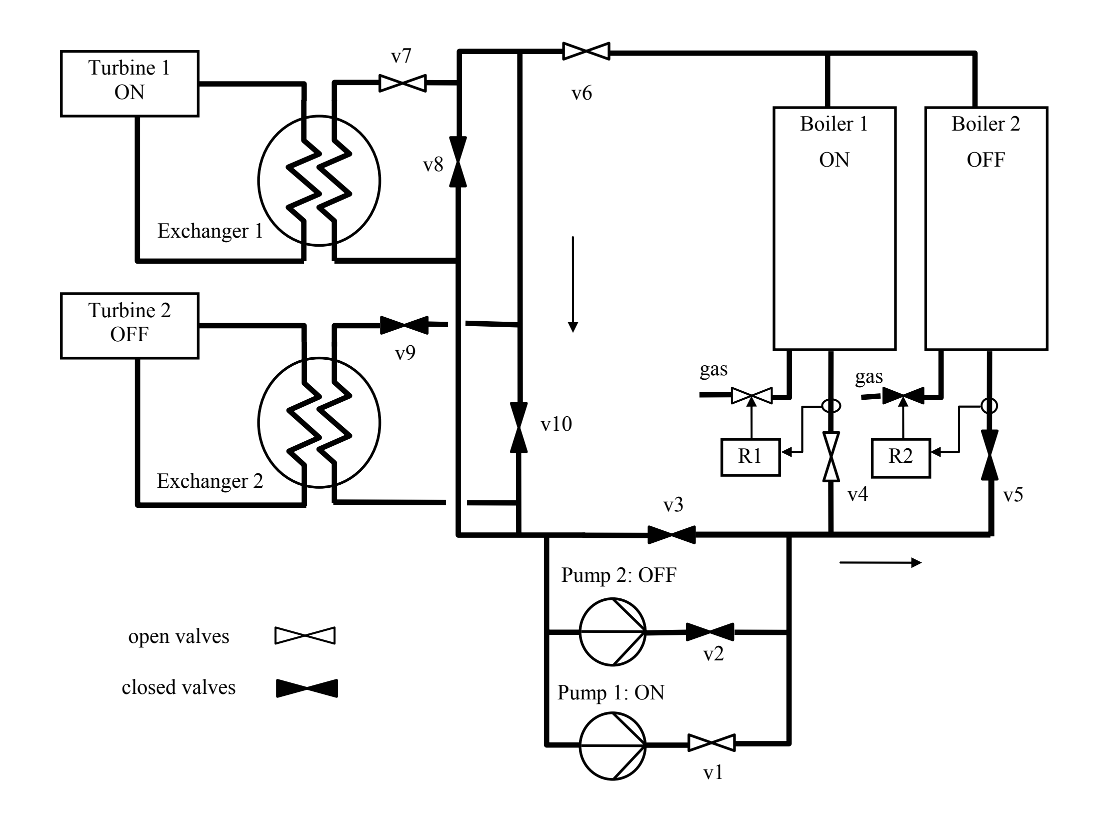


Bild 7.1 P&I-Diagramm eines einfachen Zwei-Turbinen-Kraftwerks

Die Anlage kann in drei Modi betrieben werden:

1. Geringe Leistung A: In Betrieb sind nur der Heizkessel 1, die Pumpe 1 und der Wärmetauscher 1. Die Ventile (Valves) v1, v4, v6 und v7 sind offen, alle anderen geschlossen. Das ist die Situation des P&I-Diagramms, Bild 7.1.
2. Geringe Leistung B: In Betrieb sind nur der Heizkessel 2, die Pumpe 2 und der Wärmetauscher 2. Die Ventile v2, v5, v6, v9 sind geöffnet, alle anderen geschlossen.
3. Hohe Leistung: Beide Pumpen sind in Betrieb, ebenso beide Heizkessel und beide Pumpen. Die Ventile v1, v2, v4, v5, v6, v7 und v9 sind geöffnet, alle anderen geschlossen

Es ist eine Ursachenanalyse für das *unerwünschte Ereignis* "Überhitzung des Heizkessels“ (boiler overheating) durchzuführen für den Fall, dass die Anlage gerade im Modus geringer Leistung A arbeitet. Das zu analysierende unerwünschte Ereignis ist das *Top-Ereignis* des *Fehlerbaums*, Bild 7.2 und Bild 7.3.

Zur Überhitzung des Heizkessels kann es kommen, wenn der Durchfluss durch den Heizkessel zu gering ist (low flow), und wenn gleichzeitig der Regler für die Gaszufuhr versagt und diese offen lässt (regulator failure). Diese beiden ursächlichen Ereignisse führen in konjunktiver Verknüpfung zum Top-Ereignis.

Der Reglerausfall wird als *Basisereignis* angesehen. Basisereignisse sind Ereignisse, über die man bereits genaue Kenntnisse und vor allem statistische Daten hat, und für die man nicht nach noch tiefer liegenden Ursachen suchen muss.

Anders ist das beim zu niedrigen Durchfluss. Hier sind die möglichen Ursachen noch zu ermitteln. Zum niedrigen Durchfluss kommt es, wenn die Pumpe 1 ausfällt, oder wenn die Durchströmung des Heizkessels aufgrund von Vorbeileitungen (bypass) um Pumpe oder Heizkessel herum zu gering wird, oder wenn Leitungen blockiert sind (disjunktive Verknüpfung). Damit ist die nächste Ebene des Fehlerbaums geklärt.

Diese Ursachen sind – abgesehen vom Pumpenausfall – erneut weiter aufzuhellen. Schließlich erhält man den vollen Fehlerbaum, an dessen Blättern nur Basisereignisse stehen.

Unter *Fehlerbaumanalyse* (FTA, Fault Tree Analysis) versteht man die auf den Fehlerbäumen aufbauende quantitative Analyse: Den Basisereignissen werden Wahrscheinlichkeiten zugeordnet. Über die im Fehlerbaum festgelegten Zusammenhänge wird daraus die Wahrscheinlichkeit des Top-Ereignisses ermittelt. Eines der Rechenverfahren soll in diesem Kapitel genauer betrachtet werden. Es nutzt die Indikatorfunktion. Im nächsten Kapitel sehen wir als weitere Möglichkeit die Umformung des Fehlerbaums in einen Ereignisbaum und dessen Auswertung mittels Pfadregeln.

Das Beispiel dieses Abschnitts verdanke ich einem Hinweis von Andy Dearden und Michael Harrison (1996).

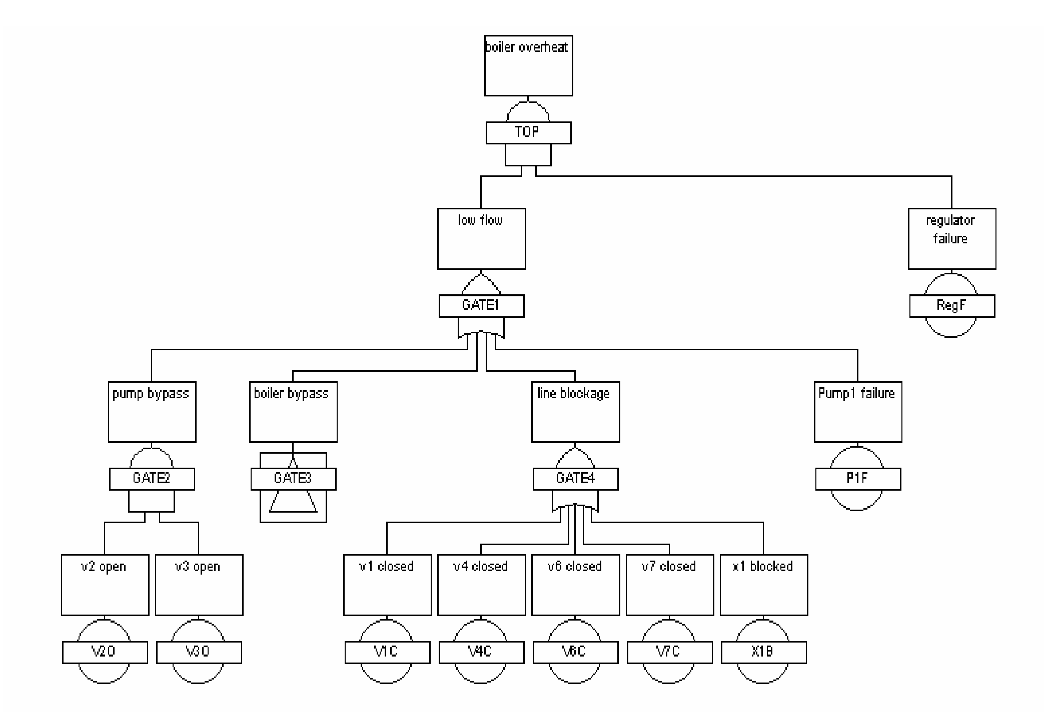


Bild 7.2 Fehlerbaum

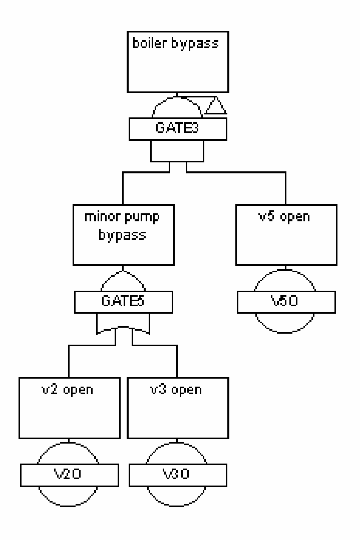


Bild 7.3 Fehlerbaum (Fortsetzung)

7.3 Indikatorfunktion und Wahrscheinlichkeiten

Der Fehlerbaum ist die grafische Darstellung einer booleschen Funktion des Top-Ereignisses in Abhängigkeit von den Basisereignissen. Diese Funktion soll nun als boolescher Ausdruck dargestellt werden.

Die Basisereignisse (das sind die Fehlerzustände der Komponenten) werden durch die booleschen Indikatorvariablen A, B, C, ... beschrieben, Tabelle 7.2. Die Variablen nehmen den Werte 1 an, wenn die entsprechende (Fehler-)Bedingung erfüllt ist, und den Wert 0 sonst.

Die Indikatorvariable I eines Ereignisses ist eine Zufallsvariable mit dem Erwartungswert $E[I] = 1 \cdot P(I{=}1) + 0 \cdot P(I{=}0) = P(I{=}1)$. Das heißt: der Erwartungswert einer Indikatorvariablen I ist gleich der Wahrscheinlichkeit des zugehörigen Ereignisses.

Die *Indikatorfunktion* für das Top-Ereignis nimmt den Wert eins an, wenn dieses Ereignis eintritt - ansonsten ist sie gleich null. Die Indikatorfunktion f_{BO} für die Überhitzung des Heizkessels ergibt sich aus dem Fehlerbaum (unter Weglassung des Operators $\wedge$ für die Konjunktion) zu

$$f_{\mathrm{BO}}(A, B, C, D, E, F, G, P, R, X) = R\,(A \vee D \vee F \vee G \vee X \vee P \vee BC \vee CE \vee BE).$$

Die gesuchte Wahrscheinlichkeit des Top-Ereignisses ist nichts anderes als der Erwartungswert dieser booleschen Funktion. Und dieser hängt offensichtlich von den Erwartungswerten der Indikatorvariablen, also von den Wahrscheinlichkeiten der Basisereignisse ab.

Für arithmetische Ausdrücke kennen wir die Regeln der Erwartungswertbildung. Daraus lassen sich die entsprechenden Regeln für die Erwartungswertbildung für boolesche Ausdrücke herleiten, beispielsweise die Regeln

$$E[a \wedge b] = E[a \cdot b] = E[a] \cdot E[b] \text{ und}$$

$$E[a \vee b] = E[a+b- a \cdot b] = E[a]+ E[b]-E[a] \cdot E[b]$$

für die Indikatorvariablen *a* und *b* zweier voneinander statistisch unabhängigen Ereignisse. Für Wahrscheinlichkeiten - bzw. Erwartungswerte - wesentlich kleiner eins kann man auch die Näherungsformel

$$E[a \vee b] \approx E[a]+ E[b] \qquad (*)$$

nehmen.

Tabelle 7.2 Die booleschen Variablen

Variable	*Bedingung (Fehlerereignis)*	*Wahrscheinlichkeit*
A	Ventil v1 geschlossen	0.001
B	Ventil v2 offen	0.01
C	Ventil v3 offen	0.01
D	Ventil v4 geschlossen	0.001
E	Ventil v5 offen	0.01
F	Ventil v6 geschlossen	0.001
G	Ventil v7 geschlossen	0.001
P	Pumpe 1 aus	0.02
R	Regler 1 (R1) offen	0.01
X	Wärmetauscher 1 blockiert	0.0001

Die exakte Ermittlung der Wahrscheinlichkeit eines Top-Ereignisses geht also folgendermaßen vor sich:

1. Ermittlung der Indikatorfunktion des Top-Ereignisses (beispielsweise mittels Fehlerbaum)
2. Ersetzung der booleschen Operatoren durch arithmetische
3. Erwartungswertbildung
4. Interpretation der Erwartungswerte als Ereigniswahrscheinlichkeiten

Für die Indikatorfunktion des Beispiels wählen wir einen etwas kürzeren Weg, indem wir uns den 2. Schritt sparen und die Näherungsformel (*) nutzen. Denn die Wahrscheinlichkeiten der Basisereignisse sind hier ja deutlich kleiner als eins. Gleichzeitig wird die Bearbeitung des Beispiels dafür genutzt, den wichtigen Begriff der *Schnittmenge* (Cut Set) einzuführen.

Wir bringen zunächst die Indikatorfunktion mit Hilfe der Gesetze der booleschen Algebra in die disjunktive Normalform:

$$f_{BO} = RA \vee RD \vee RF \vee RG \vee RX \vee RP \vee RBC \vee RCE \vee RBE$$

Die Indikatorfunktion besitzt eine negationsfreie DNF, eine DNF also, in der keine Variable in negierter Form vorkommt.

Unter einer Schnittmenge eines Fehlerbaums verstehen wir eine Menge von Ereignissen, die das Top-Ereignis zur Folge haben. Jeder einfache Term der disjunktiven Normalform definiert eine solche Schnittmenge. Sie besteht aus sämtlichen Basisereignissen der nichtnegierten Variablen dieses Terms. Treten diese Basisereignisse ein, dann wird dieser einfache Term gleich

eins, und damit auch die Indikatorfunktion. Die Ereignisse „Regler 1 offen" und „Ventil v1 geschlossen" bilden die Schnittmenge des einfachen Ausdrucks *RA*.

Unter einer *minimalen Schnittmenge* verstehen wir eine Schnittmenge, bei der kein Basisereignis weggelassen werden kann, ohne dass die Schnittmengeneigenschaft verloren geht.

Wenn die Indikatorfunktion des Top-Ereignisses eine negationsfreie DNF besitzt, wie in unserem Beispiel, dann hat das die folgenden Konsequenzen:

1. Eine Schnittmenge kann durch Hinzunahme eines weiteren Ereignisses nicht die Schnittmengeneigenschaft verlieren.
2. Man kann, ohne die Funktion zu ändern, alle diejenigen einfachen Terme weglassen, die nicht zu minimalen Schnittmengen gehören. Übrig bleiben die Terme der minimalen Schnittmengen.
3. Unter der Voraussetzung, dass die Basisereignisse voneinander statistisch unabhängig sind, und bei Anwendung der Formel (*), ist die Wahrscheinlichkeit des Top-Ereignisses näherungsweise gleich der Summe der Wahrscheinlichkeiten aller minimalen Schnittmengen. Und die Wahrscheinlichkeit jeder Schnittmenge ist gegeben durch das Produkt der Wahrscheinlichkeiten der in ihr enthaltenen Ereignisse.

Für die Wahrscheinlichkeit einer Überhitzung des Heizkessels liefert die Näherungsrechnung das Resultat $p[f_{BO}=1] = E[f_{BO}] \approx 0.000244$.

2-aus-3-System. Für ein einfaches Beispiel sollen die Wahrscheinlichkeiten einmal exakt berechnet werden. Das 2-aus-3-System besteht aus drei Subsystemen, deren Fehler voneinander statistisch unabhängig auftreten. Die Defekte der Subsysteme werden durch die Indikatorvariablen A, B und C angezeigt. Das Gesamtsystem ist genau dann defekt, wenn wenigstens zwei der Subsysteme defekt sind. Die Indikatorfunktion $f_{2\text{-aus-}3}$ eines Defekts des Gesamtsystems ist also gegeben durch $f_{2\text{-aus-}3}(A, B, C) = AB \vee AC \vee BC$. Nach Ersetzung der booleschen durch arithmetische Operatoren und nach einigen Vereinfachungen nimmt sie die Gestalt

$$f_{2\text{-aus-}3}(A, B, C) = AB+AC+BC-2ABC$$

an. Erwartungswertbildung liefert

$$\begin{aligned} E[f_{2\text{-aus-}3}(A, B, C)] &= E[AB]+ E[AC]+ E[BC]-2\,E[ABC] \\ &= E[A]E[B]+ E[A]E[C]+ E[B]E[C]-2E[A]E[B]E[C]. \end{aligned}$$

Unter der Annahme, dass jedes der Subsysteme mit derselben Fehlerwahrscheinlichkeit p defekt ist, dass also $E[A] = E[B]= E[C] = p$ gilt, ergibt sich für die Fehlerwahrscheinlichkeit $p_{2\text{-aus-}3}$ des Gesamtsystems die Formel $p_{2\text{-aus-}3} = 3p^2 - 2p^3$.

Common Cause Failure. Bis dahin haben wir nur unabhängige Fehler betrachtet. Es folgt die Darstellung einer einfachen Methode zur Behandlung von Fehlern mit gemeinsamer Ursache (*Common Cause Failure*, Andrews/Moss, 1993).

Wir betrachten ein 1-aus-2 System: Das System besteht aus zwei Komponenten. Die Fehlerzustände der Komponenten werden durch die Indikatorvariablen X_1 und X_2 beschrieben. Die Indikatorfunktion für den Systemfehler (bezogen auf die Sicherheitsspezifikation) ist $f(X_1, X_2) = X_1 \wedge X_2$.

Wir nehmen an, dass es eine gemeinsame Ursache für die Fehler der Komponenten gibt. Für die gemeinsame Ursache führen wir die Indikatorvariable C ein. Außerdem existieren für jedes der Subsysteme noch unabhängige Fehlerursachen, deren Auftreten auch unabhängig von der gemeinsamen Fehlerursache ist. Die zugehörigen Indikatorvariablen seien A für das erste und B für das zweite System. Wir setzen die Indikatorvariablen neu zusammen: $X_1 = A \vee C$ und $X_2 =$

$B \vee C$. Daraus ergibt sich für die Indikatorfunktion des Systemausfalls die Formel $f = (A \vee C) \wedge (B \vee C) = AB \vee C$.

Die Wahrscheinlichkeit des gemeinsamen Fehlers ist gegeben durch $E[C] = p_c$. Die unabhängigen Fehler mögen jeweils dieselbe Fehlerwahrscheinlichkeit besitzen: $E[A] = E[B] = p_i$.

Für die sicherheitsbezogene Fehlerwahrscheinlichkeit des Gesamtsystems ergibt sich $p_s = E[f] = E[AB \vee C] = E[AB + C - ABC] = E[A]E[B] + E[C] - E[A]E[B]E[C] = p_i^2 + p_c(1 - p_i^2)$. Für den Grenzfall, dass es keine gemeinsamen Fehlerursachen gibt, ist $p_s = p_i^2$; und falls alle Fehler auf gemeinsame Ursachen zurückgehen, gilt $p_s = p_c$.

7.4 Aussagenlogik

Die Variable A im Kraftwerksbeispiel steht nicht etwa für die Komponente, also den *Gegenstand* „Ventil v1"; sondern sie steht für den Satz oder die *Aussage* "Das Ventil v1 ist geschlossen". Sie zeigt an, ob dieser Satz wahr ist (A=1) oder eben nicht (A=0). Die booleschen Verknüpfungen dienen uns also zur Kombination von Aussagen zu komplexeren Aussagen oder Sätzen.

Nehmen wir ein Alltagsbeispiel: Wir haben die folgenden Sätze, die allesamt zutreffen mögen. Wenn es regnet, gehe ich ins Kino. Auf den Sportplatz gehe ich nur, wenn es nicht regnet. Es regnet nicht.

Wir führen für jede elementare Aussage eine Variable ein. Diesen Schritt nennen wir *Codierung*.

R: Es regnet

S: Ich gehe auf den Sportplatz

K: Ich gehe ins Kino

Das Alltagsbeispiel umfasst folgende *Prämissen*:

$R \leq K$; $S \leq \neg R$; $\neg R$

Die Prämissen sollen alle gelten, also können wir sie konjunktiv zu einer weiteren gültigen Aussage verknüpfen. Sie erfasst den gesamten Sachverhalt: $(R \leq K)(S \leq \neg R)\neg R$. Durch Anwendung der Regeln der booleschen Algebra bringen wir diesen Ausdruck in eine minimierte DNF. Es ergibt sich $\neg R$. Es bleibt also nur die Aussage übrig, dass es nicht regnet. Ob ich ins Kino oder auf den Sportplatz gehe, lässt sich aus den Prämissen nicht erschließen.

Wenn Sie zum Schluss gekommen sind, dass aus den Prämissen folgt, dass ich auf den Sportplatz gehe, dann sind sie in eine Denkfalle geraten, die später genauer untersucht wird. Tatsache ist, dass die Prämissen diesen weitreichenden Schluss *nicht* hergeben. Das Beispiel soll verdeutlichen, wie wichtig es sein kann, die Logik streng zu befolgen und gegebenenfalls die Regeln der Aussagenlogik zu Hilfe zu nehmen.

Hätte die dritte Prämisse in der Beobachtung bestanden, dass es regnet, dann hätte die *Konjunktion der Prämissen* so ausgesehen: $(R \leq K)(S \leq \neg R)R$. Und dieser Ausdruck ist äquivalent zu $K\neg SR$. Das heißt: Ich gehe ins Kino und nicht auf den Sportplatz, und es regnet.

8 Ereignisbaumanalyse

Zweck der Ereignisbaumanalyse (ETA, Event Tree Analysis) ist die qualitative und quantitative Beurteilung der Konsequenzen unerwünschter Ereignisse und die Berechnung des Risikos, das daraus folgt. Die Methode ist generell anwendbar. Voraussetzung ist, dass sich den Ereignissen (bedingte) Wahrscheinlichkeiten zuordnen lassen. Anwendungsschwerpunkt ist die Risikoanalyse von Gesamtanlagen: Mensch, Maschine (Hardware und Software) und Umwelt.

Die Analyse ist - anders als die Fehlerbaumanalyse - vorwärts gerichtet. Ausgangspunkt ist die Wurzel des Baumes; ihr wird das auslösende Ereignis zugeordnet. Den von der Wurzel ausgehenden Verzweigungen entsprechen die möglichen Folgeereignisse. Jede Verzweigung wird selbst wieder als Wurzel eines Unterbaums aufgefasst. Der Baum wird so lange verfeinert, also in Unterbäume zerlegt, bis den Verzweigungen Konsequenzen und Gewichte (Schaden, Schwere) zugeordnet werden können.

Die Fehlerbäume des vorigen Kapitels lassen sich in Ereignisbäume umformulieren. Oft findet man auch eine Kombination beider Methoden. Es gibt Programme mit grafischer Eingabe zur Risikoberechnung mit Ereignisbäumen. Sie gestatten die Integration von Fehlerbäumen in die Ereignisbäume.

8.1 Ereignisbäume

Bäume (Knuth, 1973, p. 305) eignen sich für die *Visualisierung endlicher Funktionen*. Endliche Funktionen sind Funktionen mit endlichem Definitionsbereich - und folglich auch endlichem Wertebereich.

Zur Konstruktion des Baumes einer endlichen Funktion beginnt man mit der *Wurzel* und erzeugt die Verzweigungen folgendermaßen: Der ersten Verzweigung wird eine frei gewählte unabhängige Variable zugeordnet. Jeder mögliche Wert der Variablen definiert einen eigenen *Zweig*. Jeder dieser Zweige ist die Wurzel eines Baumes, der durch die restlichen unabhängigen Variablen definiert wird. Auf diese Weise ist der gesamte Baum rekursiv definiert, vorausgesetzt, die Variablen und möglichen Werte sind vorab in eine Reihenfolge gebracht worden. Den *Blättern* des Baumes sind die Funktionswerte zugeordnet.

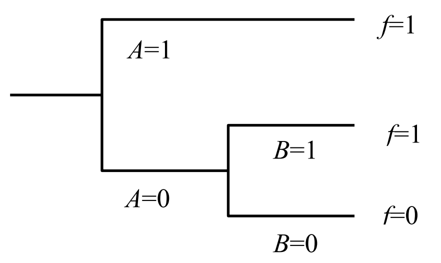


Bild 8.1 Baum der Funktion $f(A, B) = A \vee B$

Da die vollen Bäume sehr umfangreich sein können, beschneidet man sie nach folgender Regel: Unterbäume, die alle auf denselben Funktionswert führen, werden zu einem Blatt zusammengefasst. Der Funktionswert wird diesem Blatt zugeordnet.

Nehmen wir als Beispiel eine boolesche Funktion, nämlich die durch $f(A, B) = A \vee B$ definierte. Die Variablen nehmen nur die Werte 0 und 1 an, ebenso die Funktion. Bild 8.1 zeigt einen möglichen Baum dieser Funktion.

Sei X, Y, Z, ... eine endliche Kollektion diskreter (und endlicher) Zufallsvariablen. Die Zufallsvariablen können zweiwertig sein, wie beispielsweise die Indikatorvariablen, aber sie können auch einen größeren Wertebereich besitzen.

Die Funktion $f = f(X, Y, Z, ...)$ definiert dann ebenfalls eine diskrete (und endliche) Zufallsvariable. Jedem der Werte einer Zufallsvariablen entspricht ein *Ereignis*. Es wird nicht vorausgesetzt, dass die Ereignisse der verschiedenen Zufallsvariablen voneinander unabhängig sind. Für die Ereignisse und deren Kombinationen mögen die Wahrscheinlichkeiten bekannt sein.

Ein Baum der Funktion $f(X, Y, Z, ...)$ wird *Ereignisbaum* genannt. Und die darauf aufbauende Analyse heißt *Ereignisbaumanalyse*. Die Wahrscheinlichkeiten, bzw. die bedingten Wahrscheinlichkeiten, für das Durchlaufen eines Zweigs werden an die jeweiligen Baumzweige geschrieben. Die Werte der durch die Funktion f definierten Zufallsvariablen heißen *Konsequenzen.* Sie werden durch die Blätter des Baumes repräsentiert.

Die Zufallsvariable der Konsequenzen kann mit Hilfe der folgenden zwei *Pfadregeln* aus dem Ereignisbaum ermittelt werden (Sachs, 1992):

- *Multiplikationspfadregel:* Die Wahrscheinlichkeit eines Blattes ist gleich der Pfadwahrscheinlichkeit des Pfades, der von der Wurzel zu diesem Blatt führt. Die *Pfadwahrscheinlichkeit* ist gleich dem Produkt aller Wahrscheinlichkeiten der Zweige längs des Pfades.
- *Additionspfadregel:* Die Wahrscheinlichkeit eines gewissen Funktionswerts ist gegeben durch die Summe der Wahrscheinlichkeiten all derjenigen Blätter, denen dieser Funktionswert zugeordnet ist. Anders ausgedrückt: Die Wahrscheinlichkeit einer gewissen Konsequenz ist gleich der Summe der Wahrscheinlichkeiten aller Pfade, die zu derselben Konsequenz führen.

Beispiel: 1-aus-2-System. Die Indikatorvariablen für die beiden Subsysteme seien X_1 und X_2. Sie nehmen im Fehlerfall den Wert 1 an und ansonsten den Wert 0. Die Fehlerwahrscheinlichkeiten der Subsysteme werden mit p_1 und p_2 bezeichnet. Die Fehler der Subsysteme werden als statistisch unabhängig voneinander vorausgesetzt.

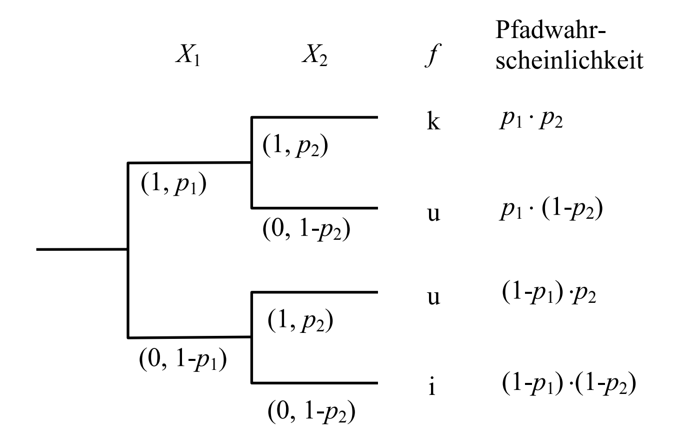


Bild 8.2 Ereignisbaum des 1-aus-2-Systems

Gemäß Sicherheitsspezifikation liegt ein Fehler nur dann vor, wenn beide Subsysteme fehlerhaft sind. Diesen Fehler bezeichnen wir als gefährlich oder kritisch. Ist nur eins der Subsysteme ausgefallen, dann handelt es sich um einen unkritischen Fehler. Wir wollen diese Fälle unterscheiden und definieren für das System die Funktion $f(X_1, X_2)$; sie zeigt die Schwere des Defekts an. Ihr Wertebereich ist {i, u, k}. Die Buchstaben stehen für „intakt", „unkritisch defekt" und „kritisch defekt".

Den Ereignisbaum der Funktion zeigt Bild 8.2. Die Variablenwerte und die Wahrscheinlichkeiten sind als Wertepaare (Variablenwert, Wahrscheinlichkeit) an den Baumzweigen notiert. Eine Verletzung der Sicherheitsspezifikation liegt vor, wenn $f(X_1, X_2)$ = k ist. Die zugehörige Fehlerwahrscheinlichkeit p_s ist gegeben durch $p_s = P(f = \text{k}) = p_1 \cdot p_2$. Eine Verletzung der funktionalen Spezifikation liegt vor, wenn $f(X_1, X_2) \in \{\text{u, k}\}$. Die zugehörige Fehlerwahrscheinlichkeit p_f ist gegeben durch $p_f = P(f \in \{\text{u, k}\}) = p_1 + p_2 - p_1 \cdot p_2$.

8.2 Bäume von booleschen Ausdrücken

Ereignisbäume können sehr umfangreich werden. Im Falle des einfachen Kraftwerks hängt die Indikatorfunktion f_{BO} von 10 unabhängigen booleschen Variablen ab. Der volle (nicht beschnittene) Baum endet in 1024 Blättern. Gefragt sind also Methoden, mit deren Hilfe man möglichst direkt zu übschaubaren beschnittenen Bäumen kommen kann. Hier wollen wir speziell Indikatorfunktionen (boolesche Funktionen) betrachten.

Den Baum einer booleschen Funktion wollen wir auf der Grundlage des *Entwicklungssatzes* konstruieren, der auf George Boole zurückgeht (Lorenzen, 1970). Die Funktion möge in Form des booleschen Ausdrucks expr vorliegen:

$$f = \text{expr}.$$

Die Variablen dieser Funktion brauchen wir zur Darstellung des Prinzips nicht näher zu bezeichnen. Wir wählen nun eine der Variablen aus; nennen wir sie v. Mit $\text{expr}^v{}_0$ bezeichnen wir den Ausdruck, der aus expr entsteht, wenn man überall v durch den Wert 0 ersetzt. Analog ist $\text{expr}^v{}_1$ bei einer Ersetzung der Variablen v durch den Wert 1 definiert. Der Entwicklungssatz hat die Form

$$\text{expr} = (v \wedge \text{expr}^v{}_1)\vee(\neg v \wedge \text{expr}^v{}_0).$$

Die Klammern sind nur zur Verdeutlichung gesetzt. Zum Beweis des Entwicklungssatzes stellt man sich eine bestimmte Wertebelegung für die Variablen vor und vergleicht, welchen Wert die linke und welchen Wert die rechte Seite der Gleichung liefert. Man sieht, dass die Erfüllungsmengen der Ausdrücke auf beiden Seiten der Gleichung identisch sind. Der Entwicklungssatz liefert unter anderem die Regel $ab\vee c = a(b\vee c)\vee\neg ac$. Nützlich ist auch die noch einfachere Formel $a\vee c = a\vee\neg ac$.

Wenn wir den Ausdruck einer Funktion einmal an die Wurzel des Baumes schreiben, lässt sich der rekursive Aufbau eines Baumes der Funktion mittels Entwicklungssatz prägnant darstellen, Bild 8.3. Wenn einer der Ausdrücke $\text{expr}^v{}_0$ oder $\text{expr}^v{}_1$ nicht mehr von Variablen abhängen, wenn er also konstant gleich 0 oder gleich 1 ist, dann ist der Zweig ein Blatt und eine weitere Entwicklung des Baumes an dieser Stelle erübrigt sich. Falls der Ausdruck aber von weiteren Variablen abhängt, kann die Entwicklung nach diesem Muster weitergehen: Der entsprechende Zweig ist Wurzel eines Baumes für diesen Ausdruck, der nun die Variable v nicht mehr enthält. Ein Beispiel ist in Bild 8.4 vollständig ausgeführt. Die zu den Zweigen gehörenden Variablen und deren Werte stehen unterhalb des jeweiligen Zweiges und der Ausdruck für den anschließenden Baum darüber.

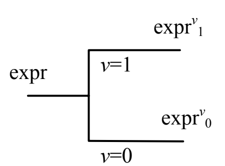


Bild 8.3 Der Entwicklungssatz

Der Umfang des Baums hängt davon ab, in welcher Reihenfolge die Variablen ausgewählt werden. Den Effekt kann man sehen, wenn man im Beispiel des Bildes 8.4 die Variablen in der Reihenfolge b, a, c wählt. Der so entstehende Baum hat ein Blatt mehr als bei der Reihenfolge a, b, c.

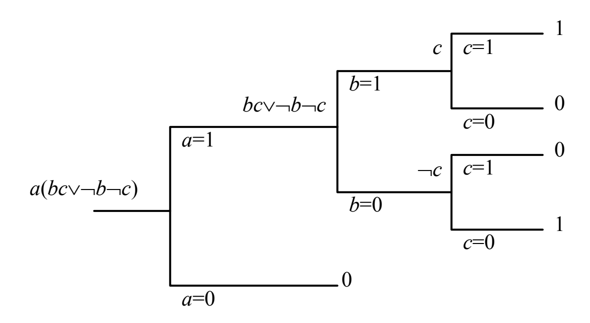


Bild 8.4 Baum des Ausdrucks $a(bc \lor \neg b \neg c)$

8.3 Risikoanalyse mit Ereignisbäumen

Den Baum der Indikatorfunktion des Kraftwerks-Beispiels zeigt das Bild 8.5. Für die Verzweigungen wurde hier folgende Konvention eingehalten: Dem Variablenwert 1 entspricht jeweils der obere, dem Variablenwert 0 der unter Zweig.

Ist der Baum eines booleschen Ausdrucks erst einmal gezeichnet, lässt er sich leicht zu einem Ereignisbaum komplettieren, indem man den Ereignissen Wahrscheinlichkeiten zuordnet. Hier sind das die Werte der Tabelle 7.2.

Der Ereignisbaum für das unerwünschte Ereignis „Überhitzung des Heizkessels" (boiler overheating) wurde mit dem Programm FaultTree+ (Item Software, 1998) gezeichnet und berechnet. Die angegebenen Wahrscheinlichkeiten Q sind hier die Wahrscheinlichkeiten für das Eintreten der Fehlerbedingung und gehören jeweils zum oberen Zweig. Mit w wird die Häufigkeit (frequency) des *auslösenden Ereignisses* bezeichnet. Hier wurde der Wert 1 gewählt. Auf diese Weise lassen sich die Häufigkeiten der letzten Spalte als Wahrscheinlichkeiten für das Durchlaufen des zugehörigen Pfades interpretieren.

Die Komponentenausfälle werden als statistisch unabhängig voneinander vorausgesetzt. Deshalb sind die Ereigniswahrscheinlichkeiten zugleich die bedingten Wahrscheinlichkeiten des zugehörigen Pfades. Die Pfadregeln ergeben für die Wahrscheinlichkeit der Heizkesselüberhitzung den Wert $p(f_{BO} = 1) = 0.000243$. Ein Vergleich mit dem Resultat des letzten Kapitels zeigt, dass bereits die Abschätzung dort ein recht genaues Ergebnis erbracht hat.

Da nur ein unerwünschtes Ereignis betrachtet wird, reicht es für die Risikobewertung aus, diese Wahrscheinlichkeit mit dem zu erwartenden Schaden bei Heizkesselüberhitzung zu multiplizieren.

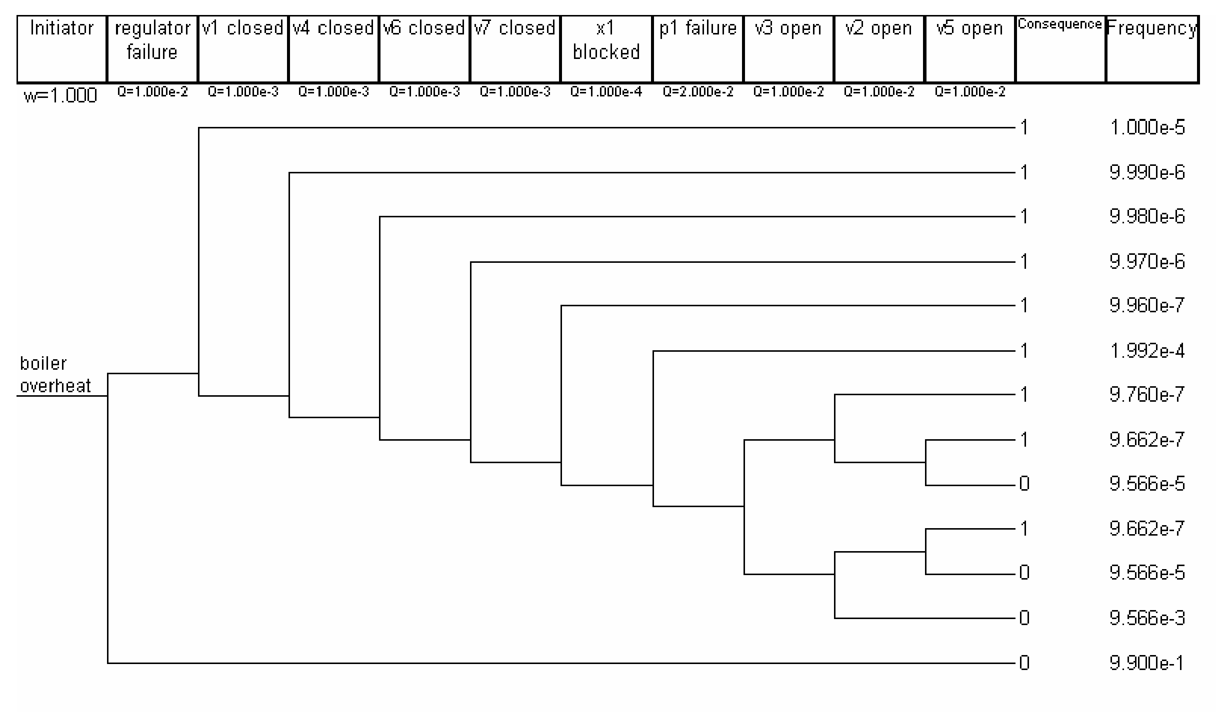


Bild 8.5 Baum der Funktion f_{BO}

9 Zuverlässigkeit komplexer Systeme

Unter einem komplexen System verstehen wir hier insbesondere ein System aus Hard- und Software. Die Zuverlässigkeitsmodellierung dieser komplexen Systeme bietet eine Fülle von Gelegenheiten für Irrtümer. Die korrekte Interpretation der Ergebnisse scheint manchmal eher Glückssache zu sein. Es gibt vielzitierte Veröffentlichungen über Zuverlässigkeitsmodelle, bei denen bereits die Modellannahmen widersprüchlich sind. Mir sind bisher schon einige Fälle von verfehlten Modellbildungen begegnet. Dazu gehören eine Formel für die Versagenswahrscheinlichkeit diversitärer Software-Systeme, die auf einer oberflächlichen Analogie zwischen Hardware- und Software-Redundanz beruht (Abschnitt 9.6) und eine zu einfache Theorie der X-Ware Reliability (Abschnitt 9.7).

Aus diesen Fehlern kann man lernen. Hier wird folgender Weg beschritten: Zunächst wird ein allgemeines Modell des Versagens komplexer Systeme formuliert. Auf dieser Basis lassen sich die Annahmen der Zuverlässigkeitsmodellierung präzise ausdrücken. Das *allgemeine Zuverlässigkeitsmodell*

- bietet Hilfestellung bei der Erstellung konkreter Zuverlässigkeitsmodelle, es
- erlaubt eine Klassifizierung der verschiedenen Arten von Zuverlässigkeitsmodellen und es
- ist Leitschnur bei der Auswahl der Auswertungs- und Analysemethoden.

Das allgemeine Zuverlässigkeitsmodell habe ich erstmals auf dem VDE-Jubiläumskongress '93 in Berlin skizziert und in einem Aufsatz (1997) in den wesentlichen Grundzügen ausgearbeitet. Es ist in den DGQ-Band 17-01 (1998) eingegangen.

9.1 Deskriptionen

Für die Zuverlässigkeitsbewertung müssen Systeme, und eventuell deren Subsysteme, nicht in ihrer vollständigen Funktion erfasst werden. Für eine abstrakte Systembeschreibung zum Zwecke der Zuverlässigkeitsbewertung wird der Begriff der *Deskription* eingeführt.

Die einfachste Deskription eines Systems ist die in der Hardware-Zuverlässigkeitsbewertung übliche Beschreibung mittels einer booleschen Variablen: Das System ist entweder intakt oder defekt. Das genügte in den Kapiteln 7 und 8.

Für komplexe Systeme ist das zu grob: Auch fehlerhafte Systeme können - mit Einschränkungen vielleicht - brauchbar sein. Und diese abgestufte Brauchbarkeit soll nun erfasst werden. Naheliegend ist, ein System durch die Menge all derjenigen Eingabedaten zu beschreiben, für die es stets korrekte Antworten liefert.

Sei Q die Deskription eines Systems. Ein Eingabedatensatz x gehört zu Q genau dann, wenn das System auf x eine Antwort liefert, und wenn alle möglichen Antworten y des Systems die Spezifikation erfüllen, wenn also $(x, y) \in R$ gilt.

Für ein deterministisches System S mit der Spezifikation R ist $Q = \mathrm{dom}(S \cap R)$. Das Korrektheitskriterium von Mills lässt sich nun so formulieren: Ein deterministisches System ist genau dann korrekt, wenn $Q = \mathrm{dom}(R)$. Zu Gunsten einer einfachen Ausdrucksweise wird in der folgenden Darstellung von deterministischen Systemen ausgegangen.

Für die Erwartungswertbildung und für logische Verknüpfungen wollen wir die Deskription noch in die Form eines Indikators bringen. Für den Indikator, dass x zur Deskription Q gehört,

nehmen wir dasselbe Symbol wie für die Deskription selbst und schreiben $Q(x)$. Es ist also $Q(x)=1$ genau dann, wenn $x \in Q$; ansonsten ist $Q(x)=0$.

9.2 Ein allgemeines Zuverlässigkeitsmodell

Aufgabe des allgemeinen Zuverlässigkeitsmodells ist es, die Zuverlässigkeitsfunktion Z durch die probabilistische Systembeschaffenheit und die Statistik des Eingabeprozesses auszudrücken.

Die Beschaffenheit des Systems - und damit seine Deskription - kann sich mit der Zeit ändern. Da sich die Systembeschaffenheit im Allgemeinen nur probabilistisch beschreiben lässt, müssen wir die zeitabhängige Deskription als stochastischen Prozess begreifen. Die probabilistische Darstellung der Beschaffenheit erfasst eingebaute Fehler, Ausfälle, Reparaturen, Hardware- und Software-Wartung. Eingabedaten und Umgebungsbedingungen werden ebenfalls probabilistisch beschrieben.

Vorausgesetzt wird, dass Eingabe und Systembeschaffenheit voneinander statistisch unabhängig sind. Die Wahrscheinlichkeit des gleichzeitigen Auftretens der Deskription Q und des Eingabedatensatzes x ist damit gleich dem Produkt der jeweiligen Wahrscheinlichkeiten.

Da wir es bei $Q(x)$ mit zwei Zufallsergebnissen zu tun haben, nämlich mit Q und mit x, muss bei der Mittelwertbildung gesagt werden, über welche der Größen gemittelt wird. Die Erwartungswertbildung schreiben wir zu diesem Zweck in der Form $E[...|Q]$ bzw. $E[...|x]$. Hinter dem senkrechten Strich steht die Variable, die festgehalten werden soll. Das ist eine Kurzschreibweise in Analogie zu den bedingten Wahrscheinlichkeiten.

Der Erwartungswert des *Versagensindikators* $1\text{-}Q(x)$ bei Mittelung über alle möglichen Eingabedaten, also $E[1\text{-}Q(x)|Q]$, ist die deskriptionsabhängige *Versagenswahrscheinlichkeit* des Systems.

Wir wollen Deskriptionen mit derselben Versagenswahrscheinlichkeit zusammenfassen. Eine derartige Zusammenfassung von Deskriptionen nennen wir *Fehlerzustand.* Die Fehlerzustände des Systems sind F_0, F_1, ..., F_n und die zugehörigen Versagenswahrscheinlichkeiten sind v_0, v_1, ..., v_n. Wir wollen annehmen, dass die Fehlerzustände nach nichtfallenden Versagenswahrscheinlichkeiten geordnet sind: $v_k \le v_{k+1}$.

Im Fehlerzustand F_0 ist die Versagenswahrscheinlichkeit gleich null: $v_0=0$. Er enthält also insbesondere die Deskription des korrekten Systems. (Der Leser möge die Namensgebung „Fehlerzustand" für etwas Korrektes verzeihen.)

Die Wahrscheinlichkeiten der Fehlerzustände können aufgrund von Ausfällen und Reparaturen zeitabhängig sein. Die zeitabhängige Wahrscheinlichkeit des Fehlerzustands F_k bezeichnen wir mit $q_k(t)$. Die *mittlere Versagenswahrscheinlichkeit* ist demzufolge im Allgemeinen auch zeitabhängig und wir bezeichnen sie mit p_t:

$$p_t = \sum_{k=1}^{n} v_k \cdot q_k(t) \ .$$

Die *Korrektheitswahrscheinlichkeit* des Systems ist mit den bisherigen Festlegungen gegeben durch $q_0(t)$. Zur Berechnung der Zuverlässigkeitsfunktion $Z(t)$ führen wir Zustände S_k ein: Der *Zustand* S_k liegt genau dann vor, wenn der Fehlerzustand gleich F_k ist und wenn *das System bisher noch nicht versagt* hat. Mit dem erstmaligen Versagen geht das System in den Zustand S_∞ über. Unter den Voraussetzungen, dass

1. die Anforderungsrate h - das ist die Anzahl der Eingabedaten je Zeiteinheit - konstant ist,

2. die Eingabe eines jeden Datums als zeitlich konzentriert angesehen werden kann und
3. das System gedächtnislos in dem Sinne ist, dass das Versagensverhalten für ein Eingabedatum nicht von den vorhergehenden Eingabedaten abhängt,

ist die *Versagensrate* $\lambda_{k\infty}$ eines Fehlerzustands F_k gegeben durch $\lambda_{k\infty} = hv_k$. Die Größe $\lambda_{k\infty}$ ist die Übergangsrate für den Übergang vom Zustand S_k in den Zustand S_∞. Mit diesen Festlegungen wird das *allgemeine Zuverlässigkeitsmodell* als abstrakter Zustandsübergangsgraph dargestellt, Bild 9.1.

Dieses Modell lässt bewusst offen, welche Übergänge zwischen den Zuständen S_0, S_1, S_2, ..., S_n überhaupt möglich sind und wie sie sich mathematisch beschreiben lassen. Die Beschreibbarkeit durch Markoff-Prozesse wird nicht vorausgesetzt.

Die *Zustände* S_k, $k \in \{0, 1, 2, ..., n, \infty\}$, treten mit den (zeitabhängigen) Wahrscheinlichkeiten $P_k(t)$ auf. $P_\infty(t)$ ist die Wahrscheinlichkeit dafür, dass das System im Zeitraum von 0 bis t wenigstens einmal versagt hat. Die Zuverlässigkeitsfunktion ist das Komplement zu eins dieser Versagenswahrscheinlichkeit

$$Z(t) = 1 - P_\infty(t).$$

Zur Zeit null hat das System noch keinmal versagt. Es ist also $Z(0) = 1$ bzw.

$$P_\infty(0) = 0.$$

Anfangs sind die Zustandswahrscheinlichkeiten gleich den Fehlerzustandswahrscheinlichkeiten:

$$P_k(0) = q_k(0) \text{ für } k = 0, 1, ..., n.$$

Die Größe $h \cdot p_t$ ist eine mittlere Versagensrate. Wir werden diese Größe nicht weiter verwenden. Jedenfalls hüte man sich vor einer voreiligen Gleichsetzung dieser mittleren Versagensrate mit der Systemversagensrate $\lambda(t) = -\dot{Z}(t)/Z(t)$. Alles hängt vom verwendeten Wahrscheinlichkeitsmaß für die Fehlerzustände und den darin berücksichtigten Bedingungen ab: Die Systemversagensrate setzt eine versagensfreie Vergangenheit voraus, die mittlere Versagensrate hingegen nicht. In Abhängigkeit von den Voraussetzungen und von der angestrebten Aussage ist es sinnvoll, einmal die mittlere Versagenswahrscheinlichkeit p_t, bei Zeitunabhängigkeit auch p, und ein andermal die Systemversagensrate $\lambda(t)$ als Kennzahl für die Systemzuverlässigkeit zu wählen.

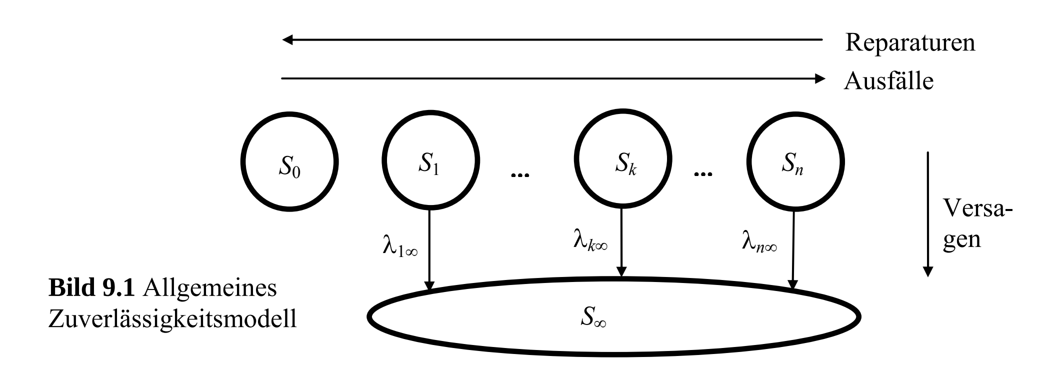


Bild 9.1 Allgemeines Zuverlässigkeitsmodell

9.3 Klassifizierung der Zuverlässigkeitsmodelle

Das allgemeine Zuverlässigkeitsmodell ist auf verschiedene Fragestellungen und Systemtypen anwendbar. Durch Spezialisierungen wird der jeweils passende Typ des Zuverlässigkeitsmodells erzeugt. Die grundlegenden Typen von Zuverlässigkeitsmodellen werden hier durch die Gegensatzpaare konstant/variant, redundant/nichtredundant und hart/weich gekennzeichnet, Tabelle 9.1. Die Tabelle 9.2 weist auf einige Modellierungsbeispiele und die zugehörigen Theorien hin.

Die weiteren Abschnitte dieses Kapitels zeigen einige dieser Spezialisierungen insbesondere für weiche Systeme, also Systeme, bei denen sich Fehler nicht immer sofort bemerkbar machen. In den folgenden Kapiteln (10 und 11) werden die Zuverlässigkeitsmodelle für einige besonders wichtige variante Systeme vertieft behandelt.

Tabelle 9.1 Modelltypen

Begriff	***Erklärung***
konstant	Der Fehlerzustand des Systems ändert sich nicht. Ausfälle und Wartung sind ausgeschlossen.
variant	Der Fehlerzustand ändert sich aufgrund von Ausfällen oder Wartungsmaßnahmen.
redundant	Nicht jeder Komponenten- oder Subsystemfehler führt zu einem Systemfehler.
nichtredundant	Jeder Komponenten- oder Subsystemfehler führt zu einem Systemfehler.
hart	Die Versagenswahrscheinlichkeiten in den Fehlerzuständen nehmen nur die Extremwerte null und eins an. Für die Versagensraten wird entweder der Wert null oder der Wert unendlich angenommen.
weich	Versagenswahrscheinlichkeiten und Versagensraten nehmen nicht nur die Extremwerte an, sondern auch Zwischenwerte.

Tabelle 9.2 Hauptarten der Zuverlässigkeitsmodellierung

	hart	***weich***
konstant	Hard- oder Software, die entweder stets versagt, oder die unveränderlich intakt ist. Ein für die Zuverlässigkeitsmodellierung uninteressanter Trivialfall.	Multi-Versionen-Programmierung. Diversitäre Programmierung (Eckhardt/Lee, 1985; Littlewood/Miller, 1987)
variant	Zuverlässigkeitsmodellierung von Hardware. Die "klassische" Zuverlässigkeitstheorie (Shooman, 1990)	Zuverlässigkeitswachstumsmodelle für Software (Lyu, 1996)

9.4 Zuverlässigkeitsmodellierung konstanter Systeme

Wir betrachten in diesem Abschnitt ein *konstantes System*, ein System also, das zwar eingebaute Fehler enthalten kann, seine Beschaffenheit aber nicht mehr ändert. Ausfall und Wartung

sind demnach ausgeschlossen. Die Wahrscheinlichkeiten der Fehlerzustände sind zeitunabhängig: $q_k(t) = q_k$. Insbesondere gilt also, dass die Korrektheitswahrscheinlichkeit q_0 konstant ist.

Programmierstudie: Orthogonal. Den 15 Teilnehmern eines Seminars wurde folgende Programmieraufgabe (Spezifikation) vorgelegt: „Gegeben sind die Winkel α und β (ganzzahlige Eingabe in °), die von zwei Geraden jeweils mit der x-Achse gebildet werden. Schreiben Sie ein Pascal-Programm zum Kopf

```
FUNCTION Orthogonal(Alpha, Beta: INTEGER): BOOLEAN;
```

Die Funktion soll genau dann den Wert TRUE liefern, wenn die beiden Geraden senkrecht aufeinander stehen."

Es wurden 15 Programmversionen angefertigt. Sie werden hier mit den Buchstaben A bis O bezeichnet. Das folgende (16.) Programm wurde zur Goldversion erklärt:

```
Orthogonal:= (Alpha-Beta) MOD 180 = 90
```

Anmerkung: Manche - nicht standardgemäße - Compiler verlangen den Text „abs(Alpha-Beta)" anstelle von „(Alpha-Beta)".

Quantitative Analyse: Per Zufallszahlengenerator wurden im Computerexperiment gleichverteilte Zufallszahlen im Bereich von -10 000 bis +10 000 für die Winkel α und β erzeugt. Das Versagen einer Programmversion wurde durch Vergleich des Ergebnisses mit dem der Goldversion festgestellt. Der Test der 15 Versionen mit 10^6 solcher Testdatenpaare lieferte das in Tabelle 9.3 dargestellte Ergebnis. Beschränkungen der Winkel auf Bereiche zwischen 0° und 360° oder zwischen -180° und +180° ergaben kein wesentlich anderes Bild.

Jede Zeile der Tabelle entspricht einer bestimmten Kombination versagender Versionen. Sie sind in der Rubrik „Beteiligte Versionen" mit einem x markiert. Die vierte Zeile der Tabelle ist beispielsweise folgendermaßen zu lesen: Exakt die zwei Versionen L und M versagen bei 55 Testdatenpaaren. Eines davon ist $(\alpha, \beta) = (-9085°, -9445°)$.

Tabelle 9.3 Testergebnis zur Orthogonal-Studie

Anzahl i zugleich versagender Versionen	Anzahl betroffener Testfälle	Beteiligte Versionen															Ein Testfall, bei dem genau i Versionen versagen	
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	α in °	β in °
0	988733																-9418	654
1	5479													x			-8823	-1983
1	89				x												3619	-3529
2	55												x	x			-9085	-9445
2	2											x	x				124	34
2	3				x									x			2295	-2025
3	28					x						x	x				7614	7524
3	1				x							x	x				-224	-314
4	34			x		x						x	x				-1352	-1262
4	23				x	x						x	x				-7125	-7215
4	1			x	x							x	x				183	273
5	22			x	x	x						x	x				9259	9349
9	1	x	x	x			x		x			x	x		x	x	609	339
10	44	x	x	x		x	x		x			x	x		x	x	-2457	-2187
10	4	x	x	x	x		x		x			x	x		x	x	227	497
11	45	x	x	x	x	x	x		x			x	x		x	x	-9528	-9798
12	111	x	x	x	x		x		x	x	x	x	x		x	x	2902	112
13	5325	x	x	x	x	x	x		x	x	x	x	x		x	x	3047	-6223

Das Zuverlässigkeitsmodell: Die Tabelle 9.3 legt die Unterscheidung von 9 Deskriptionen nahe. Sie sind in der Tabelle 9.4 zusammen mit den Versagenshäufigkeiten aufgeführt.

Die Deskriptionen Q_1 bis Q_8 besitzen alle in etwa dieselbe relative Versagenshäufigkeit von etwa 0,0055. Sie werden zu einem Fehlerzustand zusammengefasst, so dass nur noch die folgenden zwei Fehlerzustände zu unterscheiden sind:

$F_0 = \{Q_0\}$

$F_1 = \{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8\}$

Für die weitere Modellierung wird das *Urnenmodell* zu Grunde gelegt: Die tatsächlich realisierte Version wird aus den 15 Versionen des Experiments blind ausgewählt. Aus Tabelle 9.4 ergeben sich dann die Wahrscheinlichkeiten der Fehlerzustände zu q_0 = 1/15 und q_1 = 14/15. Geht man von einer Anforderung je Zeiteinheit aus (h = 1), dann ist die Versagensrate gegeben durch $\lambda_{1\infty}$ = 0,0055. Das Zuverlässigkeitsmodell wird zum Markoff-Prozess des Bildes 9.2. Die Zustandswahrscheinlichkeiten lassen sich mit den aus der Theorie der Markoff-Prozesse bekannten Methoden berechnen (Fisz, 1976).

Tabelle 9.4 Die Deskriptionen der Orthogonal-Studie

Deskription	Versionen	Versagenshäufigkeit (bei 10^6 Testfällen)
Q_0	G	0
Q_1	I, J	5436
Q_2	E	5521
Q_3	A, B, F, H, N, O	5530
Q_4	M	5537
Q_5	C	5587
Q_6	D	5624
Q_7	K	5641
Q_8	L	5696

S_0 S_1 $\lambda_{1\infty}$ S_∞

Bild 9.2 Zuverlässigkeitsmodell zur Orthogonal-Studie

Die Wahrscheinlichkeit des Zustands S_0 ist gleich der Korrektheitswahrscheinlichkeit des Systems: $P_0(t) = q_0$. Der Zustand S_1 hat die Wahrscheinlichkeit $P_1(t) = q_1 e^{-\lambda_{1\infty} t}$. Die Wahrscheinlichkeit $P_\infty(t)$ ergänzt diese beiden Wahrscheinlichkeiten auf eins. Die Zuverlässigkeitsfunktion des Systems ist gleich $Z(t) = 1 - P_\infty(t) = q_0 + q_1 e^{-\lambda_{1\infty} t}$. Sie ist - zusammen mit der Versagensrate des Systems - in Bild 9.3 zu sehen.

Obwohl es sich um ein konstantes System handelt, ist die Systemversagensrate *nicht konstant*. Ein wesentliches Merkmal des Modells ist, dass es den Software-Erstellungsprozess mit erfasst. Das Ergebnis des Prozesses - das fertige Programm - wird als zufällig angesehen. Die Berücksichtigung des Herstellungsprozesses ist unerlässlich, wenn man beispielsweise den zuverlässigkeitserhöhenden Einfluss der Software-Diversität beurteilen will.

Wir haben hier also einen ganz anderen Standpunkt eingenommen als bei der Schätzung der Zuverlässigkeit (Abschnitt 4.2). Dort haben wir einen festen aber unbekannten Fehlerzustand und eine zugehörige feste (und ebenso unbekannte) Versagensrate vorausgesetzt. Aus den Beobachtungen wurde dann ein Näherungswert für diese Versagensrate ermittelt.

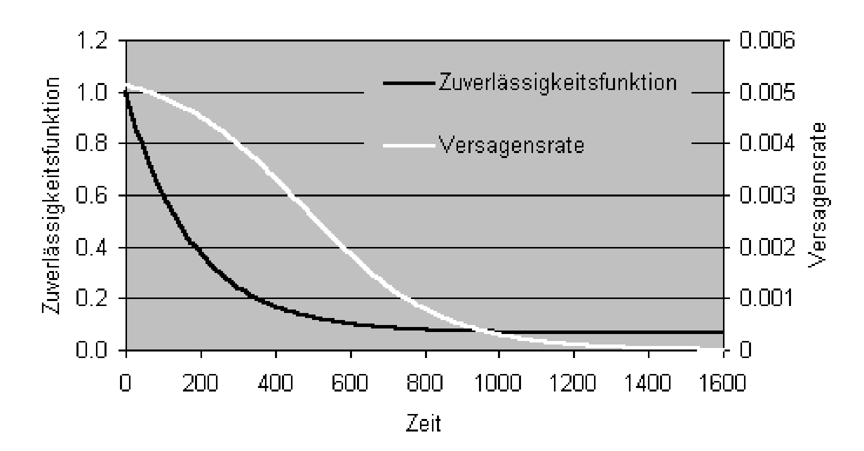


Bild 9.3 Zuverlässigkeitsfunktion und Versagensrate zur Orthogonal-Studie

9.5 Modellierung des Zuverlässigkeitswachstums

Den Zuverlässigkeitswachstumsmodellen ist ein eigenes Kapitel gewidmet, nämlich das elfte. Hier wird vorab gezeigt, wie sich diese Modelle durch Spezialisierung des allgemeinen Zuverlässigkeitsmodells gewinnen lassen. Das klärt ihre Rolle im Rahmen einer umfassenden Zuverlässigkeitsmodellierung und es macht deutlich, welchen Annahmen und Einschränkungen diese Modelle unterliegen.

Grundsätzlich geht es auch bei den Zuverlässigkeitswachstumsmodellen um eine Schätzung der Zuverlässigkeit wie in Abschnitt 4.2. Die Zuverlässigkeitswachstumsmodelle unterscheiden sich von dem dortigen Modell dadurch, dass nach jedem Versagen die Ursache ermittelt und der Fehler beseitigt wird. Danach wird das System weiter betrieben. Einer großen Klasse von *Zuverlässigkeitswachstumsmodellen* liegen die folgenden Annahmen zu Grunde.

1. Der Versagensprozess ist ein Punktprozess: Auf der Zeitachse werden nur die Versagenszeitpunkte markiert.
2. Zwischen den Versagenszeitpunkten ist der (im Allgemeinen unbekannte) Fehlerzustand konstant. Dasselbe gilt folglich für die Versagensrate.
3. Der Fehlerzustand ändert sich erst, wenn ein Versagen festgestellt und die Korrektur durchgeführt worden ist.
4. Die Fehlerlokalisierungs- und Reparaturmaßnahmen beanspruchen keine Zeit; diese Zeiten werden also aus der Zeitachse „herausgeschnitten". Anschließend wird das System weiter betrieben.
5. Für die Zeit bis zum jeweils nächsten Versagen gelten die Annahmen des allgemeinen Zuverlässigkeitsmodells mit jeweils aus dem Fehlerzustand sich ergebender konstanter (aber unbekannter) Versagensrate.

In den Zeitabschnitten zwischen den Versagenszeitpunkten wird das System demnach als konstant vorausgesetzt, es kommt nicht zu Übergängen zwischen Fehlerzuständen. Der Zeitpunkt des Übergangs von einem Zeitabschnitt zum nächsten wird durch das Versagen markiert, also durch den Übergang in den Zustand S_∞. Nur zu diesen Zeitpunkten kann sich der Fehlerzustand des Systems ändern. Die Zuverlässigkeitswachstumsmodelle sind spezielle variante Systeme.

Wir betrachten ein System, das aufgrund von Korrekturen nacheinander die Fehlerzustände F_k, F_l, F_m, ... durchläuft. Dann ist die Versagensrate bis zum ersten Versagen gleich $\lambda_{k\infty}$. Die Versagensrate in der Zeit vom ersten bis zum zweiten Versagen ist gleich $\lambda_{l\infty}$, und die Versagensrate vom zweiten bis zum dritten Versagen ist gleich $\lambda_{m\infty}$, und so weiter. Da uns die Fehlerzustände hier nicht weiter interessieren, können wir die Bezeichnung der Versagensraten vereinfachen. Wir numerieren sie einfach durch: Die Versagensrate λ_1 geht dem ersten Versagen voraus, die Versagensrate λ_2 dem zweiten, und so weiter. Der Versagensprozess lässt sich als Markoff-Prozess modellieren, Bild 9.4. Die Nummer eines Zustands ist die Zahl der bis dahin beobachteten Versagensfälle. Der Versagensprozess ist eine direkte Verallgemeinerung des Poisson-Prozesses (Abschnitt 3.6).

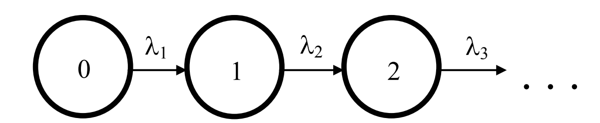


Bild 9.4 Markoff-Modell des Versagensprozesses

Anders als bei der einfachen Zuverlässigkeitsschätzung des Abschnitts 4.2 müssen wir jetzt nicht nur eine Versagensrate schätzen, sondern gleich mehrere. Und für jede der Versagensraten haben wir nur einen Beobachtungswert, nämlich den zugehörigen Versagensabstand.

Um dennoch zu statistischen Aussagen zu kommen, werden für die Versagensraten gewisse Gesetzmäßigkeiten unterstellt. Und genau diese Gesetzmäßigkeiten sind das Charakteristische eines Zuverlässigkeitswachstumsmodells. Nehmen wir als Beispiel das Jelinski-Moranda-Modell.

Die Annahmen des Jelinski-Moranda-Modells lauten: Das Programm hat eine bestimmte Anzahl n von Fehlern und jeder Fehler leistet einen Beitrag p zur Versagensrate. Anfangs ist die Versagensrate gleich np. Bei jeder Fehlerbeseitigung wird die Rate um den Anteil p reduziert. Es ist demnach $\lambda_1 = np$, $\lambda_2 = (n\text{-}1)p$, $\lambda_3 = (n\text{-}2)p$, ..., $\lambda_{n-1} = 2p$, $\lambda_n = p$.

Aufgrund der Modellannahmen sind jetzt nur noch zwei Parameter abzuschätzen, nämlich n und p. Und dafür stellt die mathematische Statistik Methoden bereit, beispielsweise die Methode der kleinsten Quadrate (Regressionsrechnung) und die Maximum-Likelihood-Schätzung (siehe Kapitel 11).

9.6 Zuverlässigkeitsmodellierung redundanter konstanter Systeme

Im allgemeinen Zuverlässigkeitsmodell findet sich die Modellbildung im Sinne der klassischen Hardwarezuverlässigkeitstheorie wieder: In diesem Fall unterscheiden die Deskriptionen nur zwischen *intakt* und *defekt*, und außerdem wird vorausgesetzt, dass im Fehlerfall die Versagensraten unendlich groß sind.

Darüber hinaus gestattet das allgemeine Zuverlässigkeitsmodell auch eine Modellierung der Zuverlässigkeit parallelredundanter Software (Multi-Versionen-Programmierung). Die klassische Zuverlässigkeitstheorie der redundanten Systeme führt bei naiver Übertragung auf die Softwarezuverlässigkeit nahezu unweigerlich zu falschen Modellen. Beispiele dafür bietet der Aufsatz „Fault-Tolerant Software Reliability Modeling“ (Scott/Gault/McAllister, IEEE Trans. Software Eng. SE-13, (1987) 5, 582-592). Durch die Theorie als auch durch empirisches Material sind diese Modelle widerlegt worden, und zwar so gründlich, dass sie auch nicht als eine „erste Näherung“ gelten können (Eckhardt/Lee, 1985; Knight/Leveson, 1985). Hier wollen wir anhand des allgemeinen Zuverlässigkeitsmodells und der Orthogonal-Programmierstudie nachvollziehen, warum.

Zwei Programmversionen, die unabhängig voneinander nach einer gemeinsamen Spezifikation entwickelt worden sind, sollen einander überwachen. Sie werden zu einem 1-aus-2-System (Kapitel 1) miteinander verbunden. Gemäß Sicherheitsspezifikation soll das gesamte System entweder die richtige Antwort liefern oder - falls die Programmversionen zu verschiedenen Ergebnissen gekommen sind - eine Fehlermeldung abgeben. Gemäß Sicherheitsspezifikation liegt ein Versagen nur dann vor, wenn beide Systeme fehlerhaft auf einen Eingabewert antworten.

Wegen der diversitären Programmierung können die Programmversionen bzw. deren Deskriptionen als *statistisch unabhängig* voneinander vorausgesetzt werden (Diversitätspostulat). Für die mittleren datenabhängigen Versagenswahrscheinlichkeiten $r_1(x)$ und $r_2(x)$ der Subsysteme und $r_{\text{div}}(x)$ des Gesamtsystems gilt dann der *Produktsatz*:

$$r_{\text{div}}(x) = r_1(x) \cdot r_2(x).$$

Beweis: Sei Q_1 die Deskription des ersten und Q_2 die des zweiten Systems. $1\text{-}Q_1(x)$ und $1\text{-}Q_2(x)$ sind die Versagensindikatoren der Subsysteme. Der Versagensindikator des diversitären Systems ist dann offenbar gleich $(1\text{-} Q_1(x))\cdot(1\text{-} Q_2(x))$. Es gilt

$$r_1(x) = E[1\text{-}Q_1(x)|x],$$

$$r_2(x) = E[1\text{-}Q_2(x)|x] \text{ und}$$

$$r_{\text{div}}(x) = E[(1\text{-} Q_1(x))\cdot(1\text{-} Q_2(x))|x].$$

Wegen der statistischen Unabhängigkeit der Deskriptionen der Subsysteme (Diversitätspostulat) ist $E[(1\text{-} Q_1(x))(1\text{-} Q_2(x))|x] = E[1\text{-} Q_1(x)|x]\cdot E[1\text{-} Q_2(x)|x]$. Und das ist gleichbedeutend mit dem Produktsatz.

Die mittleren Versagenswahrscheinlichkeiten der Teilsysteme werden mit p_1 bzw. p_2 bezeichnet; die des Gesamtsystems ist p_{div}. Es ergeben sich wegen der statistischen Unabhängigkeit von Deskriptionen und Eingabedaten die folgenden Zusammenhänge:

$$p_1 = E[1\text{-}Q_1(x)] = E[r_1(x)],$$

$$p_2 = E[1\text{-}Q_2(x)] = E[r_2(x)] \text{ und}$$

$$p_{\text{div}} = E[(1\text{-} Q_1(x))\cdot(1\text{-} Q_2(x))] = E[r_{\text{div}}(x)].$$

Der Produktsatz und die Definitionsgleichung der Kovarianz $\text{Cov}(r_1(x), r_2(x)) = E[(r_1(x)\text{-}p_1)(r_2(x)\text{-}p_2)]$ liefern nach einigen elementaren Rechenschritten die Formel von Littlewood und Miller (1987):

$$p_{\text{div}} = p_1\, p_2 + \text{Cov}(r_1(x), r_2(x)).$$

Diese Formel erfasst sowohl den Fall der erzwungenen Diversität als auch den der zufälligen Diversität. Erzwungene Diversität zeichnet sich gegenüber der zufälligen Diversität durch zusätzliche Vorgaben aus. Diese Vorgaben sind für die Programmierer der ersten Version anders als für die Programmierer der zweiten Version. Dadurch sollen Programmversionen entstehen, die wesentlich voneinander abweichen (Saglietti/Ehrenberger/Kersken, 1992).

Bei zufälliger Diversität ergeben sich einfachere Formeln. Dann sind nämlich die Versagenswahrscheinlichkeiten der Subsysteme gleich: $r_1(x) = r_2(x) = r(x)$ bzw. $p_1 = p_2 = p$. Die Kovarianz wird zur Streuung σ^2 der datenabhängigen Versagenswahrscheinlichkeit. Die Formel von Littlewood und Miller geht dann über in die Formel von Eckhardt und Lee (1985):

$$p_{\text{div}} = p^2 + \sigma^2.$$

Eine direkte Übertragung des Zuverlässigkeitsmodells von der Hardware auf die Software liefert demgegenüber die Formel $p_{\text{div}} = p^2$. Sie vernachlässigt die Tatsache, dass das Versagens-

verhalten der beiden Subsysteme über die gemeinsamen Eingabedaten miteinander verkoppelt ist.

Was würde Diversität im Falle der Orthogonal-Programmierstudie bringen? Da es sich um eine recht kleine Stichprobe handelt, muss die Formel von Eckhardt/Lee auf ein *Urnenmodell ohne Zurücklegen* übertragen werden: Sei K die Zahl der verfügbaren Versionen (Stichprobengröße), und i die Anzahl der Versionen, die beim Datensatz x ausfallen. Die datenabhängige Versagenswahrscheinlichkeit des einfachen Systems ergibt sich damit zu $r(x) = i/K$ und die des diversitären Systems zu $r_{\text{div}}(x) = \frac{i(i-1)}{K(K-1)}$. Bei der Orthogonal-Studie ist $K = 15$ und i findet man für die verschiedenen Testdatensätze in der ersten Spalte der Tabelle 9.3. Die Werte für p und p_{div} ergeben sich durch Mittelwertbildung über alle Testdatensätze x. Hier die Ergebnisse: $p \approx 0{,}005$, $p_{\text{div}} \approx 0{,}004$ und $p_{\text{div}}/p^2 \approx 160$.

Das heißt, dass die Formel $p_{\text{div}} = p^2$ auch nicht als „erste Näherung" taugt. Die Streuung wird zum dominierenden Teil und man muss in vielen Fällen mit der wesentlich ungünstigeren Näherungsformel $p_{\text{div}} \approx \sigma^2$ rechnen. Und das Schlimme ist, dass man σ normalerweise gar nicht kennt. Im Falle der Orthogonalstudie fallen die Subsysteme meist zusammen aus und die Versagenswahrscheinlichkeit des diversitären Systems ist nahezu genauso groß wie die des einfachen.

Im Falle der Orthogonal-Studie bringt Diversität also gar nichts. Gründe für das Scheitern der Diversität sind

- *Konzentrationseffekte:* Fehler konzentrieren sich auf bestimmte „kritische" Datensätze. Kritische Datensätze der Orthogonal-Studie sind Winkeldifferenzen von Vielfachen von 90 Grad. Im Schnitt versagt etwa jedes zweite Programm bei kritischen Datensätzen. Das erklärt die Versagenswahrscheinlichkeit von 0.0055.
- *Denkfallen*: Sie sind dafür verantwortlich, dass die Fehlerwahrscheinlichkeiten für kritische Datensätze groß sind. Eine Denkfalle der Orthogonal-Studie manifestiert sich in der Deskription Q_3. Typisch ist folgendes Programm (siehe auch Abschnitt 14.3):

```
IF (Alpha+90=Beta) OR (Alpha-90=Beta) THEN
   Orthogonal:= true
ELSE Orthogonal:= false
```

Konzentrationseffekte und Denkfallen sorgen für eine große Schwankung der datenabhängigen Versagenswahrscheinlichkeit $r(x)$ des Einzelsystems. Und diese Schwankungen finden ihren Ausdruck in einer großen Standardabweichung σ.

9.7 Einfache variante Systeme: X-Ware Reliability

Variante Systeme ändern ihre Beschaffenheit. Das heißt, dass es zu Übergängen zwischen den Fehlerzuständen kommen kann. Solche Systeme sind außerordentlich schwer zu fassen. Von praktischer Bedeutung sind zwei Sonderfälle. Der erste wird im Rahmen der Zuverlässigkeitswachstumsmodelle behandelt; der zweite dient hier als Gegenbeispiel gegen eine zu einfache Theorie der X-Ware Reliability.

Laprie und Kanoun (Lyu, 1996) betrachten Systeme aus Hard- und Software, in denen es zu Ausfällen und daraufhin zu Versagen kommen kann. Sie nennen die Systeme X-Ware. Ihre Theorie der X-Ware Reliability ist der Versuch, die Hardware-Zuverlässigkeit und die Software-Zuverlässigkeit zu kombinieren und in einem einheitlichen Theoriegebäude unterzubringen, wie das ja auch hier angestrebt wird. Aber das ist noch nicht alles: Meines Erachtens ver-

suchen Sie darüber hinaus zu zeigen, dass sowohl das Ausfallverhalten als auch das Versagensverhalten und schließlich auch noch die Überlagerung beider sich jeweils durch die Exponentialverteilung beschreiben lässt.

Schauen wir uns das System genauer an: Es kann ausfallen und es wird nicht repariert. Ferner ergibt sich nach einem Ausfall eine ganz bestimmte Ausfallrate. Das allgemeine Zuverlässigkeitsmodell kommt unter diesen Voraussetzungen mit drei Zuständen aus: S_0, S_1 und S_∞. Die Ausfallrate wird mit λ_{01} und die Versagensrate mit $\lambda_{1\infty}$ bezeichnet, Bild 9.5.

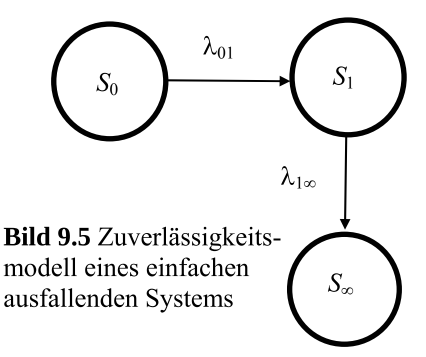


Bild 9.5 Zuverlässigkeitsmodell eines einfachen ausfallenden Systems

Die Zuverlässigkeitsfunktion dieses Systems ist gegeben durch

$$Z(t) = (\lambda_{1\infty} e^{-\lambda_{01} \cdot t} - \lambda_{01} e^{-\lambda_{1\infty} \cdot t})/(\lambda_{1\infty}-\lambda_{01}).$$

Die Herleitung wird im 10. Kapitel nachgeholt. Das Markoff-Modell entspricht nämlich, abgesehen von Umbenennungen der Übergangsraten, genau dem des Bildes 10.2.

Die Versagensrate des Systems $\lambda(t)$ ist anfangs null, sie wächst monoton und nähert sich einem Endwert. Dieser Endwert ist gleich der kleineren der beiden Übergangsraten. Bild 9.6 zeigt die Verläufe. Durch geeignete Normierung der Zeitachse wurde die kleinere der beiden Übergangsraten gleich 1 gesetzt. Zu sehen sind die Zeitverläufe der Systemversagensrate für verschiedene Werte der größeren der Übergangsraten. Die Zahlen der Kurven geben ihren jeweiligen Wert an.

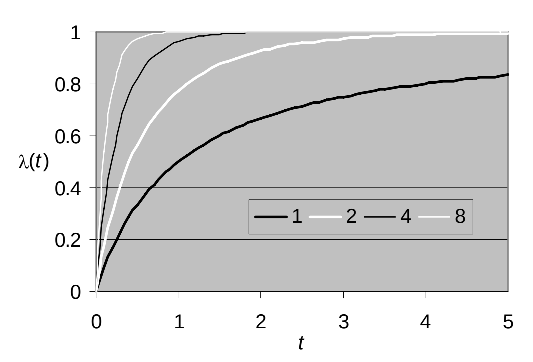


Bild 9.6 Die Systemversagensrate

Mit einer konstanten Versagensrate kann man näherungsweise nur dann rechnen, wenn die größere der beiden Raten die kleinere wesentlich übersteigt. Dann geht die Zuverlässigkeitsfunktion entweder über in $e^{-\lambda_{1\infty} \cdot t}$ oder sie wird zu $e^{-\lambda_{01} \cdot t}$. Jedenfalls bestimmt bei stark unterschiedlichen Übergangsraten immer die kleinere der beiden das dynamische Verhalten. Die größere der Raten sorgt dafür, dass der durch sie bestimmte Anteil rasch an Einfluss verliert.

Eine konstante Versagensrate erhält man also nur in einem der beiden folgenden Fälle:

1. Die Ausfallrate ist so groß, dass (gemessen an der Zeit bis zum Versagen) ein Ausfall praktisch sofort eintritt. Das System reagiert, als sei es von Anbeginn defekt. Die Zustände S_0 und S_1 verschmelzen miteinander. Das ist genau der Fall, der in der *Theorie der Software-Zuverlässigkeit* untersucht wird.
2. Nach einem Ausfall ist sofort mit einem Versagen zu rechnen. Hier verschmelzen die Zustände S_1 und S_∞ miteinander. Genau dieser Fall ist Gegenstand der klassischen *Hardware-Zuverlässigkeitstheorie*.

Insgesamt ergibt sich also die ernüchternde Schlussfolgerung, dass die Suche nach einer einfachen Zuverlässigkeitstheorie komplexer Systeme mit konstanten Versagensraten erfolglos bleiben wird. In einer Theorie der Zuverlässigkeit komplexer Systeme kommen konstante Ver-

sagensraten nur in den Grenzfällen vor, die bereits in den speziellen Zuverlässigkeitstheorien für Hardware oder Software ihren Platz hatten.

9.8 Ermittlung der Korrektheitswahrscheinlichkeit

Das am wenigsten zugängliche Maß der Zuverlässigkeitsbewertung ist die Korrektheitswahrscheinlichkeit k: Die Korrektheit lässt sich durch Tests nicht feststellen - wie aber dann? Eine Möglichkeit ist, die Korrektheit zu beweisen. Dann gilt k=1 und jeder ist zufrieden. Was aber, wenn ein solcher Beweis aus nachvollziehbaren Gründen nicht zu haben ist? Bleibt dann nur noch die Wahrsagerei?

Es folgen ein paar unkonventionelle Vorschläge zur Ermittlung der Korrektheitswahrscheinlichkeit.

1. Der *Wettquotient* als Schätzwert für Korrektheitswahrscheinlichkeit: Man könnte die Kenner eines Systems (Entwickler, Konstrukteure, Tester, Projektleiter usw.) sowie die Kenner der Firmenorganisation und -kultur auf die Korrektheit wetten lassen. Gewettet wird auf eine im Nachhinein prüfbare Aussage, beispielsweise: „Innerhalb der nächsten zwei Jahre wird es kein Versagen des Systems geben". Diese Aussage wird mit der Korrektheit gleichgesetzt. Nehmen wir an, Person A setzt einen Betrag a auf die Richtigkeit der Aussage und B den Betrag b dagegen; k ist die Korrektheitswahrscheinlichkeit. Die Gewinnerwartung für A ist gleich kb-(1-k)a und für B gleich $-kb$+(1-k)a. Bei Chancengleichheit ist die Gewinnerwartung beider gleich null. Daraus folgt $k = a/(a+b)$. Also: Unter der Bedingung der Chancengleichheit ist der Wettquotient $a/(a+b)$ ein Maß für die Korrektheitswahrscheinlichkeit (Carnap/Stegmüller, 1959; Sachs, 1992).
2. Die *negative Methode:* Die Ermittlung des Wettquotienten und auch die Feststellung des Bewährungsgrades (siehe weiter unten) kann nur in einer Unternehmenskultur funktionieren, die auf Offenheit und auf die negative Methode setzt. Die Fehler der Vergangenheit sind ein teuer erworbener Schatz. Für die Zukunft können Fehler aber nur dann ihren Wert entwickeln, wenn sie erstens erkannt und zweitens vielen bekannt werden. Elemente der negativen Methode sind die gnadenlose Suche nach Fehlern und der offene Umgang mit ihnen. Derjenige, der selbstfabrizierte Fehler mitteilt, muss belohnt werden!
3. *Entscheidung bei Risiko:* Die Entscheidungslehre (Kapitel 12) ist auf die obige Wettsituation anwendbar. Die Entscheidungsalternativen heißen in diesem Fall: 1. Verzicht auf die Wette, 2. Wetten auf Korrektheit und 3. Wetten auf Unkorrektheit. Die Entscheidungslehre verspricht Antworten auf die Fragen: Welche Rollen spielen Risikoaversion und Risikosympathie? Ist die Korrektheitswahrscheinlichkeit wirklich gleich dem Wettquotienten? Wie ist die Formel gegebenenfalls zu korrigieren? Was ist zu tun, damit die Wette überhaupt zustande kommt und aussagekräftige Ergebnisse liefert?
4. *Bewährungsgrad:* Wenn es nicht gelingt, die Korrektheitswahrscheinlichkeit abzuschätzen, sollte man ein etwas bescheideneres Ziel anpeilen und mit einer Abschätzung des Bewährungsgrads zufrieden sein: Sei T ein korrekt entworfener Test. Das heißt: Jedes nach der gegebenen Spezifikation korrekt entworfene System möge den Test bestehen. Mit P_T wird die Wahrscheinlichkeit bezeichnet, dass das System diesen Test besteht. Ein Test ist um so wirksamer, je kleiner P_T ist. Sei k die Korrektheitswahrscheinlichkeit vor dem Test und k_T die Korrektheitswahrscheinlichkeit nach Bestehen des Tests. Das Verhältnis k_T/k nennen wir *Bewährungsgrad* durch den Test T. Die Regeln der Wahrscheinlichkeitsrechnung liefern die Formel $\frac{k_T}{k} = \frac{1}{P_T}$ (Abschnitt 2.1). Wenn ein System den Test nur mit der

Wahrscheinlichkeit von 10% besteht, dann steigt die Korrektheitswahrscheinlichkeit durch den bestandenen Test um den Faktor zehn. Auch für die Ermittlung von P_T muss man auf Schätzverfahren und Erfahrungswissen zurückgreifen. Jedenfalls ist es leichter, etwas über die Wirksamkeit eines Tests und damit über den Zuwachs der Korrektheitswahrscheinlichkeiten (k_T/k) zu erfahren, als über die Korrektheitswahrscheinlichkeiten selbst (k_T oder k). Aggressive Tests zeichnen die negative Methode aus. Ziel ist eine möglichst große Wirksamkeit, also ein möglichst kleines P_T.

10 Wartung und Reparatur: Verfügbarkeit

In diesem Kapitel werden nur harte variante Systeme betrachtet. Das ist klassische Zuverlässigkeitstheorie für Hardware. Die Zuverlässigkeitsfunktion $Z(t)$ ist die Wahrscheinlichkeit dafür, dass bis zum Zeitpunkt t das System nicht ausfällt, kurz: die Überlebenswahrscheinlichkeit.

10.1 Das Problem

Fehlertoleranz kann wirkungslos sein. Nehmen wir ein 2-aus-3-System. Die Zuverlässigkeitsfunktion einer jeden der Komponenten sei gegeben durch

$$Z(t) = e^{-\lambda t} .$$

Die Fehlerwahrscheinlichkeit p eines Subsystems zur Zeit t ist also gegeben durch

$$p = 1 - Z(t) = 1 - e^{-\lambda t} .$$

Dementsprechend ist die Fehlerwahrscheinlichkeit des Gesamtsystems zu diesem Zeitpunkt gegeben durch

$$p_{2\text{-aus-}3} = 1 - Z_{2\text{-aus-}3}(t).$$

Die Formel $p_{2\text{-aus-}3} = 3p^2 - 2p^3$ aus Abschnitt 7.3 wird damit zu

$$1 - Z_{2\text{-aus-}3}(t) = 3(1 - e^{-\lambda t})^2 - 2(1 - e^{-\lambda t})^3 .$$

Daraus folgt für die Zuverlässigkeitsfunktion des Gesamtsystems

$$Z_{2\text{-aus-}3}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t} .$$

Die mittleren Zeiten bis zum Ausfall für ein Subsystem und für das Gesamtsystem ergeben sich daraus zu

$$MTTF = 1/\lambda \text{ und}$$

$$MTTF_{2\text{-aus-}3} = \frac{5}{6\lambda} = \frac{5}{6} MTTF .$$

Die Formel wird auch durch folgende einfache Überlegung plausibel: Die mittlere Zeit bis zu dem Zeitpunkt, zu dem eins der drei Subsysteme ausfällt, beträgt $MTTF/3$, wie man sich mit Hilfe der Ausfallratenaddition klar machen kann. Bis ein weiteres Subsystem und damit das Gesamtsystem ausfällt, vergeht im Mittel die Zeit $MTTF/2$. Addiert man diese beiden Werte, kommt man auf das obige Ergebnis.

Das verblüffende Ergebnis ist: Die *MTTF* des redundanten Systems ist geringer als diejenige einer Komponente.

Im Hinblick auf die bloße *MTTF* als Kennzahl bringt die Parallelredundanz demnach gar nichts. Bereits im Kapitel über Sicherheitstechnik wurde darauf hingewiesen, dass Fehlertoleranz erst dann richtig wirksam wird, wenn Fehler rechtzeitig beseitigt werden. Dieses Kapitel bietet die rechnerischen Grundlagen für die Festlegung optimaler Reparatur- und Wartungsstrategien.

10.2 Redundante Systeme bei periodischer Wartung

Einen Eindruck davon, wann Parallelredundanz profitabel sein kann, vermittelt ein Vergleich der Zuverlässigkeitsfunktionen von Subsystem und Gesamtsystem, Bild 10.1.

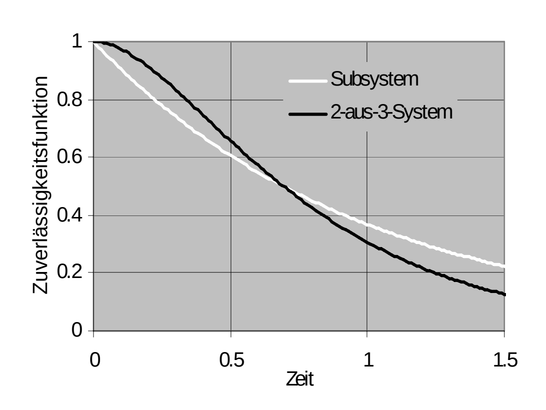


Bild 10.1 Zuverlässigkeitsfunktionen ($\lambda = 1$)

Für Zeiten unterhalb von 69 % der *MTTF* eines Subsystems liegt die Zuverlässigkeitsfunktion des 2-aus-3-Systems oberhalb derjenigen der Subsysteme. Sie hat im Zeitnullpunkt eine horizontale Tangente. Für Systeme mit einer Missionsdauer deutlich unterhalb der *MTTF* zahlt sich Parallelredundanz demnach aus.

Bei größeren Betriebsdauern kann man diesen Effekt ebenfalls nutzen, wenn man das System periodisch wartet. Die Wartungsintervalle mögen die Dauer T haben. Außerdem sei das Wartungsintervall wesentlich kleiner als die *MTTF*: $T \ll MTTF$. Bei jedem Wartungsvorgang werden alle Fehler beseitigt und das System wieder in den ursprünglichen Zustand versetzt. Für relativ kleine Wartungsintervalle T dürfen Näherungen erster und zweiter Ordnung für die Zuverlässigkeitsfunktionen verwendet werden. Es ergeben sich folgende Näherungen

$$Z(t) \approx 1 - \lambda t \text{ bzw.}$$

$$Z_{\text{2-aus-3}}(t) \approx 1 - 3(\lambda t)^2.$$

für $\lambda t \leq \lambda T \ll 1$. Die Fehlerwahrscheinlichkeiten sind also gleich λt für das Subsystem und gleich $3(\lambda t)^2$ für das gesamte 2-aus-3-System. Die Fehlerwahrscheinlichkeiten verringern sich unter den genannten Bedingungen also um den Faktor $3\lambda t$. Die Redundanz ist umso wirkungsvoller, je kleiner λT ist. Diese Zahl ist gleich dem Verhältnis aus Wartungsintervall und *MTTF* eines Subsystems: $T/MTTF$.

Eine geeignete Zuverlässigkeitskennzahl für gewartete Systeme ist die *mittlere Verfügbarkeit A* (Availability). Dieser Wert ist der Zeitmittelwert der Zuverlässigkeitsfunktion über das Wartungsintervall. Es ergibt sich

$$A = \frac{1}{T}\int_0^T Z(t)\mathrm{d}t \approx 1 - \lambda T/2 \text{ bzw.}$$

$$A_{\text{2-aus-3}} = \frac{1}{T}\int_0^T Z_{\text{2-aus-3}}(t)\mathrm{d}t \approx 1 - (\lambda T)^2.$$

Die mittleren Unverfügbarkeiten sind $U = 1 - A = \lambda T/2$ für die Komponenten und $U_{\text{2-aus-3}} = 1 - A_{\text{2-aus-3}} = (\lambda T)^2$ für das System. Auch deren Verhältnis sagt etwas über die Wirksamkeit der Redundanz in Verbund mit der Wartungsstrategie aus: $U_{\text{2-aus-3}}/U = 2\lambda T$. Die Kennzahl weist aus, dass die Parallelredundanz eine erhebliche Zuverlässigkeitssteigerung ergibt, wenn das Wartungsintervall relativ klein ist.

Die hier vorgenommene Quantifizierung der Wirksamkeit von Redundanz zusammen mit der Wartungsstrategie ermöglicht eine gezielte Festlegung der Wartungsintervalle, wie das beispielsweise in der Sicherheitstechnik gefordert wird.

Grundsätzlich gilt das bisher Gesagte für alle einfehlertoleranten Systeme. Wir verallgemeinern nun die Betrachtungen auf Systeme aus n Komponenten. Alle Komponenten mögen dieselbe Ausfallrate λ besitzen. Das System sei einfehlertolerant. Also, erst bei Ausfall einer zweiten Komponente kommt es zum Systemfehler. Das Markoff-Modell des Bildes 10.2 erfasst die Verhältnisse. Im Zustand 0 ist keine Komponente defekt. Im Zustand 1 ist genau eine Komponente defekt und im Zustand 2 sind es zwei. Die Übergangsraten ergeben sich zu $\lambda_1 = n\lambda$ und $\lambda_2 = (n\text{-}1)\lambda$.

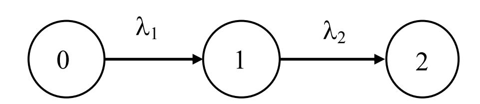


Bild 10.2 Markoff-Modell eines Ausfallprozesses

Mit $p_i(t)$ wird die Wahrscheinlichkeit des Zustands i bezeichnet. Anfangs ist das System intakt: $p_0(0) = 1$, $p_1(0) = 0$, $p_2(0) = 0$. Die Zustandswahrscheinlichkeiten erfüllen die Differentialgleichungen

$$\dot{p}_0(t) = -\lambda_1 p_0(t),$$

$$\dot{p}_1(t) = \lambda_1 p_0(t) - \lambda_2 p_1(t),$$

$$\dot{p}_2(t) = \lambda_2 p_1(t).$$

Die Lösung dieses Anfangswertproblems ist gegeben durch

$$p_0(t) = e^{-\lambda_1 t},$$

$$p_1(t) = \frac{\lambda_1}{\lambda_2 - \lambda_1}(e^{-\lambda_1 t} - e^{-\lambda_2 t}),$$

$$p_2(t) = 1 - \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 t}.$$

Daraus ergibt sich die Zuverlässigkeitsfunktion des Systems $Z_s(t)$ zu

$$Z_s(t) = 1 - p_2(t) = \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 t} - \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 t}.$$

Für das einfehlertolerante System heißt das

$$Z_s(t) = n\, e^{-(n-1)\lambda t} - (n-1)\, e^{-n\lambda t}.$$

Die mittlere System-Lebensdauer ist gleich

$$MTTF_s = \frac{2n-1}{n(n-1)\lambda} = \frac{2n-1}{n(n-1)} MTTF.$$

Hierin ist $MTTF$ wieder die mittlere Lebensdauer einer einzelnen Komponente. Für $n = 3$ ergeben sich, wie zu erwarten ist, die Formeln für das 2-aus-3-System. Die Zuverlässigkeitserhöhung durch Wartung lässt sich ganz analog dem obigen Muster ermitteln.

10.3 Reparierbare Systeme

Wenn ein System über Fehlererkennungsmechanismen (Diagnose) verfügt, kann nach Eintritt eines Fehlers die Reparatur beginnen. Der Ausfallprozess besteht dann aus einer abwechselnden Folge von Zeiten, in denen das System ausgefallen ist und solchen, in denen das System intakt ist. Diese Zeiten sind zufällig. Die Länge eines Intervalles, in dem das System intakt ist, nennen wir auch *Ausfallabstand*, Bild 10.3.

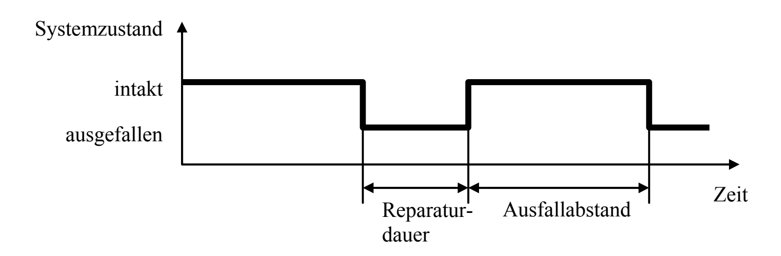


Bild 10.3 Der Ausfallprozess eines reparierbaren Systems

Wir führen für den mittleren Ausfallabstand die Kennzahl *MTBF* (Mean Time Between Failures) ein. Die *MTBF* ist anders definiert als die *MTTF*. Falls bei Reparaturen jeweils der ursprüngliche Zustand wieder hergestellt wird („Es ist wieder wie neu"), unterscheiden sich die Zahlenwerte von *MTBF* und *MTTF* nicht.

Der Mittelwert der *Reparaturdauern* heißt *MTTR* (Mean Time To Repair). Die Wahrscheinlichkeit dafür, rein zufällig auf ein intaktes System zu treffen, heißt mittlere *Verfügbarkeit A*. Und die Wahrscheinlichkeit, rein zufällig auf ein ausgefallenes System zu treffen, ist die mittlere *Unverfügbarkeit U.* Zwischen den Größen bestehen folgende einfach einzusehende Zusammenhänge:

$$A = \frac{MTBF}{MTBF + MTTR} \text{ bzw. } U = \frac{MTTR}{MTBF + MTTR}\,. \qquad (*)$$

Wir wollen nun stationäre Prozesse betrachten, also Prozesse, bei denen die Wahrscheinlichkeiten der einzelnen Zustände mit der Zeit gegen feste Werte streben. In einem solchen Fall ist die *stationäre Verfügbarkeit* gleich der mittleren Verfügbarkeit; Entsprechendes gilt für die Unverfügbarkeit.

Nun soll wieder das einfehlertolerante System aus *n* gleich zuverlässigen Komponenten (Ausfallrate λ) betrachtet werden. Mit den bereits oben eingeführten Bezeichnungen $\lambda_1 = n\lambda$ und $\lambda_2 = (n\text{-}1)\lambda$ ergibt sich für den Ausfallprozess das Markoff-Modell des Bildes 10.2.

Das System sei reparierbar, so dass die Fehlerzustände wieder verlassen werden können. Die Reparatur wird durch die Übergangsraten μ_1 und μ_2 beschrieben. Das Modell geht über in das des Bildes 10.4.

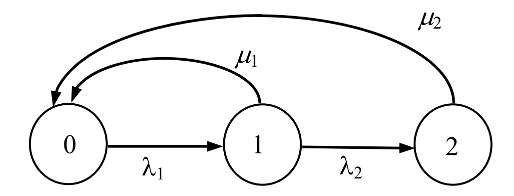


Bild 10.4 Markoff-Modell eines Ausfall- und Reparaturprozesses

Nun soll eine *stationäre Analyse* dieses Prozesses erfolgen. Wir gehen also davon aus, dass die Wahrscheinlichkeiten der Zustände sich ihrem stationären Wert angenähert haben, so dass die zeitlichen Ableitungen der Zustandswahrscheinlichkeiten gleich null sind. Für die stationäre Wahrscheinlichkeit des Zustands mit der Nummer i schreiben wir p_i. Zwischen den Zustandswahrscheinlichkeiten bestehen folgende Beziehungen:

$$- \lambda_1 p_0 + \mu_1 p_1 + \mu_2 p_2 = 0$$

$$\lambda_1 p_0 - (\mu_1 + \lambda_2) p_1 = 0$$

$$\lambda_2 p_1 - \mu_2 p_2 = 0$$

$$p_0 + p_1 + p_2 = 1$$

Die ersten drei Gleichungen sind voneinander linear abhängig, so dass man eine davon auch weglassen kann. Insgesamt hat man drei Gleichungen für die drei Unbekannten. Für p_2 erhält man die Lösung

$$p_2 = \frac{1}{1 + \frac{\mu_2}{\lambda_1} + \frac{\mu_2}{\lambda_2} + \frac{\mu_1}{\lambda_1} \cdot \frac{\mu_2}{\lambda_2}} .$$

Da das System einfehlertolerant ist, ist es nur im Zustand 2 tatsächlich ausgefallen. Dieser Zustand wird mit der Rate μ_2 verlassen. Die mittlere Dauer dieses Ausfallzustands ist $1/\mu_2$, das ist die mittlere Reparaturdauer bei Systemausfall. Die Wahrscheinlichkeit p_2 dieses Zustands ist gleichzeitig die Unverfügbarkeit des Systems. Über die sinngemäße Anwendung der Formeln (*) kann man jetzt den mittleren Ausfallabstand $MTBF_s$ des Systems berechnen. Die Formel liefert $p_2 = 1/(\mu_2 \cdot MTBF_s + 1)$ und daraus ergibt sich

$$MTBF_s = \frac{1}{\lambda_1} \cdot (1 + \frac{\lambda_1 + \mu_1}{\lambda_2}) .$$

Nun substituiert man die Übergangsraten durch die Vielfachen der Komponentenausfallrate gemäß $\lambda_1 = n\lambda$ und $\lambda_2 = (n\text{-}1)\lambda$. Es ergibt sich

$$MTBF_s = \frac{2n - 1 + \frac{\mu_1}{\lambda}}{n(n-1)\lambda} .$$

Ohne Reparatur ($\mu_1 = 0$) erhält man die $MTTF_s$ des nicht gewarteten und nicht reparierten Systems, die bereits oben errechnet wurde.

Die Reparaturrate μ_2 ist nur als Rechengröße eingeführt worden, um überhaupt die stationäre Analyse durchführen zu können. In der Formel für die $MTBF_s$ kommt sie nicht mehr vor. Die eigentlich wichtige Größe ist die Reparaturrate μ_1 für die Beseitigung des ersten Fehlers. Unter der Annahme, dass diese Reparaturrate wesentlich größer als die Ausfallrate einer Komponente ist, kann man zu folgender Näherung übergehen:

$$MTBF_s \approx \frac{1}{n\lambda} \cdot \frac{\mu_1}{(n-1)\lambda} .$$

Da $1/(n\lambda)$ der mittlere Ausfallabstand des nichtredundanten Systems ist, ergibt sich durch die Reparatur eine Verbesserung des mittleren Ausfallabstands um den Faktor $\frac{n}{n-1} \cdot \frac{\mu_1}{n\lambda}$. Dieser Faktor ist in etwa gleich dem Verhältnis des mittleren Ausfallabstands des nichtredundanten Systems $1/(n\lambda)$ und der mittleren Reparaturdauer $1/\mu_1$.

11 Zuverlässigkeitswachstumsmodelle

Bei den Zuverlässigkeitswachstumsmodellen geht man davon aus, dass nach jedem Versagen die Fehlerursache ermittelt und beseitigt wird. Das heißt, dass die Zufallsvariablen T_1, T_2, ... für die Versagenszwischenzeiten zwar immer noch als exponentialverteilt angesehen werden können, dass aber die Parameter der Verteilung λ_1, λ_2, ... verschieden sein können. Zuverlässigkeitswachstumsmodelle machen gewisse Annahmen über die Beziehungen zwischen diesen Parametern. In der Literatur sind etwa achtzig solcher Modelle bekannt.

Ursprünglicher Zweck der Zuverlässigkeitswachstumsmodelle ist die *Prognose* der zukünftig zu erwartenden Zuverlässigkeit eines Systems aus den Daten der Vergangenheit. Also: Aus den bis zum Zeitpunkt t bekannten Versagenszwischenzeiten t_1, t_2, t_3, ..., t_n (das sind Realisierungen der Zufallsvariablen T_1, T_2, T_3, ..., T_n) wird auf die zukünftige Versagensrate bzw. deren zeitlichen Verlauf geschlossen.

Hinsichtlich der Zeit wollen wir in diesem Kapitel grundsätzlich voraussetzen, dass nur die reine Betriebszeit gemessen wird; das heißt, dass die Uhr angehalten wird, sobald sich das System nicht im Einsatz oder im Testbetrieb befindet.

Der Zweck der Prognose wird mit den Zuverlässigkeitswachstumsmodellen meines Erachtens nicht erreicht. Im letzten Abschnitt begründe ich diesen Standpunkt. Dennoch sind Zuverlässigkeitswachstumsmodelle nicht zwecklos. Ich sehe ihre eigentliche Stärke darin, dass sie es ermöglichen, die Wirksamkeit zuverlässigkeitserhöhender Maßnahmen im Nachhinein zu beurteilen. Nicht die Prognose, sondern die *Retrospektive* halte ich für das eigentliche Feld der Zuverlässigkeitswachstumsmodelle. Und so verstanden können sie ein wirkungsvolles Mittel im Rahmen der *negativen Methode* sein: Ein ungünstiger Verlauf der Versagensraten spricht gegen das aktuell betriebene Zuverlässigkeitsmanagement und legt eine Änderung nahe.

Eine umfassende Darstellung von Zuverlässigkeitswachstumsmodellen enthält das Buch von Lyu (1996). Zum Buch erhält der Käufer eine CD-ROM mit einer Fülle von Fehlerstatistiken aus Software-Projekten und Programmen.

Hier konzentriere ich mich auf die einfachsten Modelle der Zuverlässigkeitsprognose, und zwar solche, die sich auch für die *Retrospektive* eignen.

11.1 Naive Zuverlässigkeitsschätzung

Wer das Wetter für morgen nach dem einfachen Rezept voraussagt, dass sich gegenüber dem heutigen Wetter wohl wenig ändern wird, liegt meist ziemlich richtig. Diese Wettervorhersage wird von den Meteorologen als *naive Prognose* bezeichnet. In Analogie zur naiven Wettervorhersage formuliere ich die *naive Zuverlässigkeitsschätzung*.

Das Modell der naiven Zuverlässigkeitsschätzung bzw. -prognose basiert auf der Annahme, dass ein Zuverlässigkeitswachstum nicht vorliegt, oder zumindest so gering ist, dass es in der Vorhersageungenauigkeit untergeht. Das naive Modell geht also von konstanten und gleichen Versagensraten für die in Betracht zu ziehenden letzten n zufälligen Versagenszwischenzeiten T_1, T_2, ..., T_n aus und macht Voraussagen über die zukünftigen Versagenszwischenzeiten T_{n+1}, T_{n+2}, ..., T_{n+k}, das heißt, es liefert nach der Methode aus Abschnitt 4.2 einen Schätzwert n/t für den Parameter λ dieser Verteilungen und das Vertrauensintervall. Hierbei ist $t = t_1+t_2+ ... +t_n$ die Summe der n zuletzt festgestellten Versagensabstände. Der Parameter n wird vorab festge-

legt. Wenn über den Produktionsprozess und das Produkt nichts weiter bekannt ist, wollen wir $n = 5$ setzen.

Wählt man die Stichprobenanzahl derartig klein, dann muss man mit großen Schwankungen des Vorhersagewertes im Laufe des Fehlerbeseitigungsprozesses rechnen. Dafür ist der Erwartungswert dieser Schätzung nahe am tatsächlichen Wert.

Zur Verringerung der Schwankung und Verkleinerung des Vertrauensintervalls kann man die Stichprobe vergrößern. Aber je größer die Stichprobe, desto weniger wird das Zuverlässigkeitswachstum durch den Schätzwert gewürdigt.

Bei sehr komplexen Produkten und bei geringem Zuverlässigkeitsanspruch ist in späten Testphasen auch eine größere Stichprobe gerechtfertigt (beispielsweise n=10). Das lässt sich so begründen: Vermutlich sind dann noch viele Fehler im System. Aber alle diese Fehler haben eine sehr kleine Fehleroffenbarungswahrscheinlichkeit, so dass die Beseitigung eines Fehlers die Versagensrate des Systems nicht wesentlich verändert.

11.2 Das Modell von Duane

Erfahrungen, die man mit Produkten macht, führen zu Korrekturen und diese wiederum zu besseren Produkten. Es ist also kein Wunder, wenn ein Produkt im Laufe der Zeit immer besser wird. Demnach nehmen die Versagensraten der produzierten Systeme mit der Zeit ab (Birolini, 1991, S. 280 ff.; O´Connor, 1991, S. 308 ff.). Duane entwickelte daraus ein empirisches Gesetz.

Der *kumulierte mittlere Ausfallabstand* eines Systems, das nach Ausfall jeweils durch ein verbessertes Exemplar ersetzt wird, ist definiert als die gesamte verstrichene Zeit t geteilt durch die bis dahin beobachtete Anzahl von Ausfällen n. Er ist also gleich t/n. Trägt man diese Werte im doppelt logarithmischen Maßstab über der Zeit auf, erhält man in vielen Fällen näherungsweise eine Gerade. Deren Steigung wird mit α bezeichnet. Die Gerade stellt folgenden Zusammenhang zwischen dem kumulierten Ausfallabstand und der Anzahl der beobachteten Ausfälle her:

$$\frac{t}{n} = \frac{t_0}{n_0}\left(\frac{t}{t_0}\right)^{\alpha}.$$

Zum Zeitpunkt t_0 wird die Anfangsphase mit bis dahin n_0 Ausfällen abgeschlossen. Die darauffolgende Zeit ist der Bereich, auf den sich die Vorhersage bezieht.

Der Erwartungswert des Ausfallabstands zur Zeit t lässt sich durch $\frac{\mathrm{d}t}{\mathrm{d}n}$ ausdrücken. Aus der Formel von Duane ergibt sich dafür der Wert

$$\frac{\mathrm{d}t}{\mathrm{d}n} = \frac{1}{1-\alpha}\cdot\frac{t_0}{n_0}\cdot\left(\frac{t}{t_0}\right)^{\alpha} = \frac{1}{1-\alpha}\cdot\frac{t}{n}.$$

Also: Der Erwartungswert des Ausfallabstands zur Zeit t ist gleich dem kumulierten mittleren Ausfallabstand bis dahin geteilt durch $1-\alpha$. Der Kehrwert des Erwartungswerts des Ausfallabstands ist ein Schätzwert für die momentane Ausfallrate.

Das einfache Duane-Modell lässt sich auch auf den Prozess der Software-Wartung anwenden: Entdeckte Fehler werden beseitigt; an die Stelle des Ausfalls tritt das Versagen.

Der Parameter α hängt vom Qualitätsmanagement des Unternehmens ab und davon, welche Anstrengungen zur Zuverlässigkeitsverbesserung unternommen werden. Als Richtschnur für die Wahl des Parameters möge die Tabelle 11.1 dienen. Für $\alpha = 0$ haben wir es mit einer Varianten der naiven Zuverlässigkeitsschätzung zu tun (Birolini, 1991, S. 439-442).

Letztlich läuft das gesamte Modell von Duane auf nichts anderes hinaus als eine Anwendung der naiven Prognose n/t für die Versagensrate und einer Multiplikation des Ergebnisses mit einem - irgendwie begründeten - "Optimismusfaktor" $1-\alpha$.

Beispiel: Eine vorläufige Zuverlässigkeitsbewertung eines elektronischen Systems ergab 11 Ausfälle in 600 Stunden. Für den endgültigen Betrieb wird ein mittlerer Ausfallabstand von wenigstens 500 Stunden gefordert. Angesichts der zuverlässigkeitserhöhenden Maßnahmen möge für den Parameter des Duane-Modells ein Wert von $\alpha = 0.5$ gerechtfertig sein. Daraus lässt sich eine Abschätzung für die Dauer der noch notwendigen Nachbesserungen gewinnen. Die Formel des Duane-Modells ergibt eine erforderliche Betriebsdauer von insgesamt t = 12 604 h. Davon ist die bisher verstrichene Zeit von 600 h abzuziehen. Für die Nachbesserungsphase sind also noch wenigstens 12 000 Stunden einzuplanen.

Tabelle 11.1 Anhaltspunkte für die Wahl des Parameters im Duane-Modell

α	***Zuverlässigkeitserhöhende Maßnahmen***
0	Keinerlei zuverlässigkeitserhöhende Maßnahmen. Angenommen wird eine konstante Versagensrate. Einfache Zuverlässigkeitsschätzung entsprechend der naiven Zuverlässigkeitsprognose.
< 0.2	Es werden keine besonderen Anstrengungen zur Zuverlässigkeitserhöhung unternommen. Fehler werden nicht analysiert. Korrekturen nur bei erheblichen Fehlerauswirkungen, aber mit geringem Nachdruck.
0.2	Zuverlässigkeitserhöhung genießt normale Aufmerksamkeit. Tests werden ohne besondere Intensität durchgeführt. Fehler mit erheblichen Auswirkungen werden korrigiert.
0.3 ... 0.4	Zuverlässigkeitswachstum genießt hohe Priorität. Tests werden mit erhöhter Intensität durchgeführt. Die Analyse und Beseitigung von Fehlern sind gut organisiert.
0.4 ... 0.6	Die Fehlerbeseitigung hat höchste Priorität. Effiziente Teststrategien werden eingesetzt. Fehler werden sofort analysiert. Die Korrekturmaßnamen sind hochwirksam.

11.3 Das geometrische Modell

Das Duane-Modell ist eine empirische Kurve, deren mathematische Formel allein vom vorhandenen Datenmaterial bestimmt wurde und die keiner weiteren theoretischen Begründung bedurfte. Die Zuverlässigkeitswachstumsmodelle (RGM, Reliability Growth Model) für Software beruhen auf einem etwas anderen Zugang: Es wird eine Gesetzmäßigkeit für das Zuverlässigkeitswachstum der Software vorausgesetzt. Dieses Modell hat ein paar Parameter, mit denen es an die vorhandenen Daten angepasst werden kann. Die Zuverlässigkeitsbewertung und -prognose geschieht also in zwei Schritten: Zuerst ist ein passendes Modell zu wählen, und dann sind die Parameter dieses Modells aufgrund des vorhandenen Datenmaterials abzuschätzen (Birolini, 1991/1999; Lyu, 1996; Shooman, 1990).

Für die Parameterschätzung greift man auf die Standardverfahren der mathematischen Statistik zurück (Fisz, 1976). Insbesondere sind das die Maximum-Likelihood-Schätzung und die Parameterschätzung nach der Methode der kleinsten Quadrate (Least-squares estimators).

Die Modellierung geschieht unter gewissen Annahmen. Einige dieser Annahmen sind einer großen Klasse von Zuverlässigkeitswachstumsmodellen gemeinsam (Kapitel 9.5):

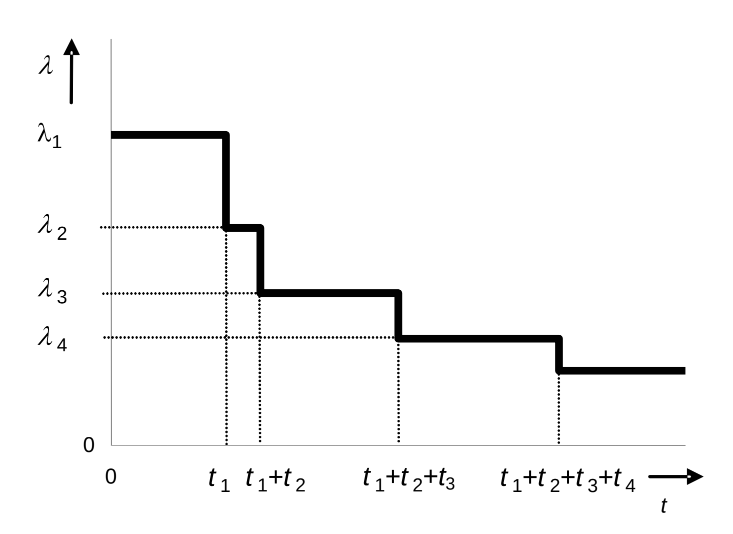
1. Die Versagensrate hängt nur vom zwar unbekannten aber als fest angenommenen Fehlerzustand des Systems ab.
2. Zu Änderungen des Systems und dessen Fehlerzustands kommt es nur dann, wenn ein Versagen festgestellt worden ist. Daraus folgt:
3. Die Versagenszwischenzeiten sind exponentialverteilt.
4. Bei einem erkannten Fehler kommt es sofort zu einer Korrekturmaßnahme. Über den Erfolg der Korrekturmaßnahme wird zunächst nichts weiter vorausgesetzt. Es kann zum Einbau neuer Fehler kommen.
5. Um den Versagensprozess als reinen Punktprozess darstellen zu können, wird nur die Betriebszeit berücksichtigt. Die Zeit für Fehlerlokalisierung und -korrektur wird aus der Zeitachse "herausgeschnitten".
6. Die Versagenszwischenzeiten sind voneinander statistisch unabhängig. Das heißt unter anderem, dass Operationsprofil und Testanstrengungen *nicht* von der Versagensstatistik der Vergangenheit abhängig gemacht werden.
7. Für Prognosen ist zusätzlich zu fordern, dass die Daten unter Bedingungen erhoben werden, wie sie auch im späteren Betrieb gelten.

Die Versagensrate des Systems vor Beseitigung des i-ten Fehlers ist λ_i. Die tatsächlich verstrichene Zeit vom i-1-ten bis zum i-ten Fehler wird mit t_i bezeichnet. Diese Zeiten heißen auch *Versagensabstände*. Es mögen n Fehler beobachtet und beseitigt worden sein. Die Zeiten t_1, t_2, t_3, ..., t_n sind also bekannt. Sie sind Realisierungen der exponentialverteilten Zufallsvariablen T_1, T_2, T_3, ..., T_n. Für den Erwartungswert der exponentialverteilten Zufallsgröße T_i gilt $E[T_i] = 1/\lambda_i$.

Beim *geometrischen Zuverlässigkeitswachstumsmodell* kommt zu den obigen allgemeinen Annahmen noch die folgende hinzu (Lyu, 1996):

8. Die Versagensraten nehmen geometrisch ab (Bild 11.1). Es gilt also $\lambda_i/\lambda_{i-1} = p$ mit einer Konstanten p und für alle i = 2, 3, 4, ..., n. Demzufolge nehmen die Erwartungswerte der Versagensabstände mit steigendem i geometrisch zu.

Da diese Modellannahmen einmal von P. B. Moranda vorgeschlagen worden sind, spricht man auch vom Zuverlässigkeitswachstumsmodell von Moranda.



11.4 Maximum-Likelihood-Schätzung

Wir bleiben bei den Annahmen des geometrischen Modells von Moranda. Die Versagensraten mögen also nach jeder Fehlerbeseitigung um den Faktor p schrumpfen. Die Versagensrate bis zum ersten Versagen bezeichnen wir mit d. Also ist die Versagensrate zwischen dem i-1-ten und dem i-ten Versagen gleich $\lambda_i=d\cdot p^{i-1}$.

Um zu Schätzwerten für die Parameter d und p zu kommen, gehen wir der Frage nach, für welche Parameter p und d, die beobachtete Folge von Versagensabständen t_1, t_2, t_3, ..., t_n die größte Wahrscheinlichkeit hat. Mit der Wahrscheinlichkeit $\lambda_i \cdot e^{-\lambda_i \cdot t_i}\,\mathrm{d}t$ fällt der beobachtete Wert t_i tatsächlich in ein um t_i herum gelegenes Intervall der Größe dt. Das gilt für alle i. Da die Werte unabhängig voneinander sind, erhält man die Gesamtwahrscheinlichkeit als Produkt der Einzelwahrscheinlichkeiten. Durch diese Überlegungen ist die sogenannte Likelihood-Funktion motiviert. Unter Weglassung der nicht weiter interessanten Konstanten dt erhält man sie zu

$$L(p, d) = \prod_{i=1}^{n} \lambda_i \cdot e^{-\lambda_i \cdot t_i} = d^n \prod_{i=1}^{n} p^{i-1} \cdot e^{-d \cdot p^{i-1} \cdot t_i} \;.$$

Diejenigen Werte der Modellparameter p, d, die diese Funktion maximal machen, sind deren Maximum-Likelihood-Schätzwerte.

Die Ermittlung dieser Schätzwerte ist hier etwas leichter, wenn wir zum Logarithmus der Funktion übergehen. Das ist gefahrlos möglich, denn die Werte p und d, die $L(p, d)$ maximieren, sind dieselben, die $\ln(L(p, d))$ maximal machen. Die Werte ergeben sich aus den beiden Gleichungen

$$d = \frac{p \cdot n}{\sum_{i=1}^{n} p^i \cdot t_i} \quad \text{und}$$

$$\frac{\sum_{i=1}^{n} i \cdot p^i \cdot t_i}{\sum_{i=1}^{n} p^i \cdot t_i} = \frac{n+1}{2} \,.$$

Aus der zweiten Gleichung wird mit irgendeinem numerischen Verfahren zur Nullstellenbestimmung (zum Beispiel einer einfachen Intervallschachtelung) der Parameterwert p ermittelt. Einsetzen in die erste der Gleichungen liefert d.

11.5 Regressionsrechnung

Wir legen die Modellannahmen des geometrischen Modells zugrunde. Wenn wir die natürlichen Logarithmen der Zeiten bis zum Ausfall bilden

$$x_i = \ln(t_i)$$

und diese über die Fehlernummern i auftragen, dann wird sich diese Punktwolke aufgrund der Modellannahmen gut durch eine Gerade der Form

$$f(i) = a \cdot i + b$$

annähern lassen. Jedenfalls liegen die Logarithmen der Erwartungswerte $E[T_i]$ auf einer solchen Geraden.

Wir verzichten hier darauf, die Parameter d und p des geometrischen Modells direkt zu ermitteln. Stattdessen werden die Parameter a und b so bestimmt, dass die Summe der quadratischen Abweichungen der Punkte x_i von der Geraden minimal wird:

$$\sum_{i=1}^{n}(x_i-(a\cdot i+b))^2 \to \min .$$

Die Minimierung liefert für die Parameter a und b die Formeln

$$a=\frac{(\sum_i i\cdot x_i)/n-\bar{x}\cdot\bar{i}}{(\sum_i i^2)/n-\bar{i}^2}=\frac{\sum_i i\cdot x_i-\bar{x}\cdot\bar{i}\cdot n}{\sum_i (i-\bar{i})^2} \quad \text{und}$$

$$b=\bar{x}-a\cdot\bar{i}\,. \tag{*}$$

Hierin wurden für die Mittelwerte der i- und der x-Werte die Abkürzungen

$$\bar{i}=\frac{1}{n}\sum_{i=1}^{n} i=\frac{n+1}{2}$$

und

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n} x_i$$

verwendet. Aufgrund der Gleichung (*) lässt sich die Regressionsgerade in eine anschaulichere Form bringen:

$$f(i)=\bar{x}+a\cdot(i-\bar{i})\,.$$

Wie lässt sich die Regressionsgerade interpretieren? Der Einfachheit halber wollen wir zur Veranschaulichung des Ergebnisses einmal davon ausgehen, dass die Bedingungen des naiven Modells gültig sind. Also: Es möge keine Zuverlässigkeitserhöhung vorliegen, was gleichbedeutend mit der Annahme $p=1$ ist. Die Werte $t_1, t_2, t_3, ..., t_n$ können alle als Realisierung einer exponentialverteilten Zufallsvariabeln T mit dem Parameter $\lambda = d$ aufgefasst werden. Unter diesen Voraussetzungen ist a näherungsweise null und die Regressionsgerade wird zu einer konstanten Funktion $f(i) \approx \bar{x} \approx E[\ln(T)]$. Was wir wissen wollen, ist der Parameter λ oder auch dessen Kehrwert E[T]. Es besteht folgender Zusammenhang:

$$E[\ln(T)]=\int_0^\infty \ln(t)\cdot\lambda\cdot e^{-\lambda t}\,\mathrm{d}t=\int_0^\infty(\ln(\lambda t)-\ln(\lambda))e^{-\lambda t}\lambda\mathrm{d}t=-C-\ln(\lambda)=-C+\ln(1/\lambda).$$

Mit C wird die Eulersche Konstante bezeichnet. Sie hat den Wert $C = 0.577215665$. Für den mittleren Versagensabstand erhalten wir damit $1/\lambda = e^{C+E[\ln(T)]} = e^C e^{E[\ln(T)]} = 1.781\ e^{E[\ln(T)]}$. Um einen Schätzwert für $1/\lambda$ zu bekommen, müssen wir demnach auf den aus der Regressionsgeraden abzulesenden Schätzwert $f(i)$ für $E[\ln(T)]$ die Exponentialfunktion anwenden und das Ergebnis mit dem Korrekturfaktor 1.781 multiplizieren.

11.6 Anwendungsbeispiel

Bild 11.2 zeigt für die letzten 50 Werte des Datensatzes SYS1 von Musa (Lyu, 1996) das Ergebnis der Zuverlässigkeitsmodellierung mit den hier behandelten Modellen und Methoden. Mit der naiven Schätzung ergibt sich für den letzten Schritt die Versagensrate zu etwa 1/2495s. Die Parameterschätzung nach der Maximum-Likelihood-Methode ergibt den Wert 1/1898s. Einen pessimistischeren Schätzwert liefert das Modell von Duane, nämlich 1/1769s. Und mit der Regressionsrechnung kommt man im letzten Schritt auf einen noch höheren Wert, nämlich 1/1526s.

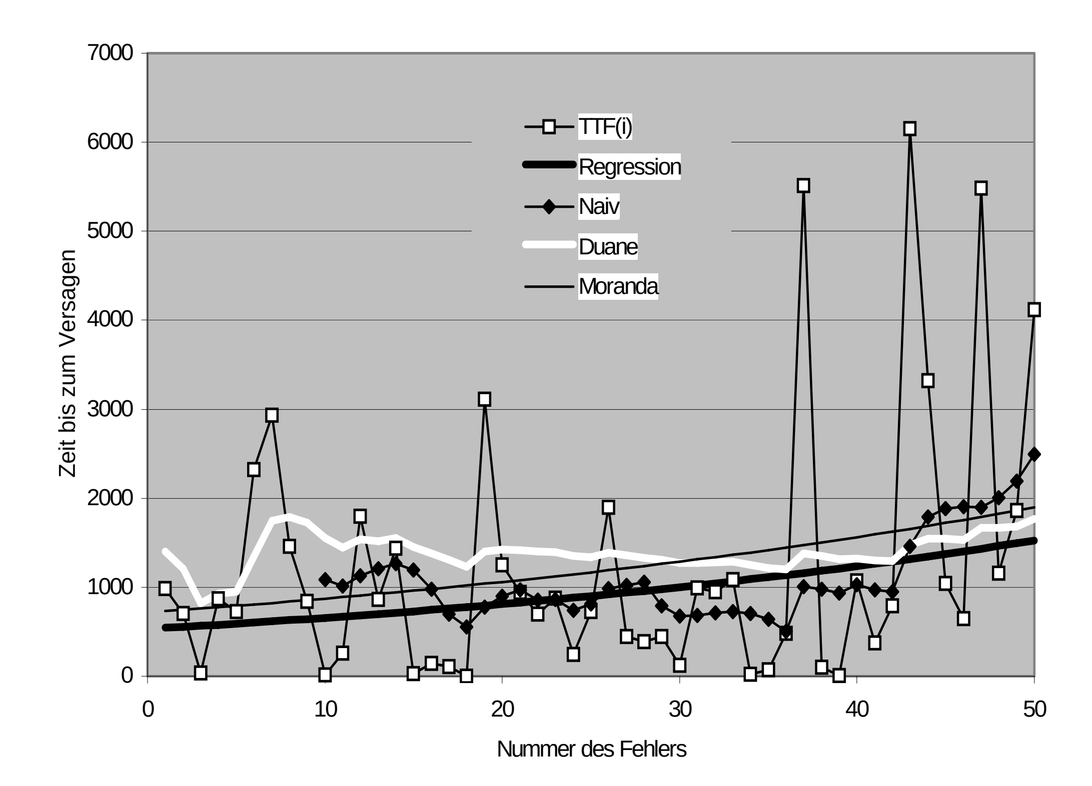


Bild 11.2 Musa-Daten, Abschätzungen der mittleren Zeit bis zum Versagen mittels Regression, mit der naiven Prognose (basierend auf den jeweils letzten 10 Werten), nach Duane ($\alpha = 0.3$) und mit dem geometrischen Modell von Moranda. Mit TTF(i) werden die Versagensabstände t_i bezeichnet (Time To Failure).

11.7 Was zeichnet nützliche Prognosen oder Theorien aus?

Um den Nutzen von Zuverlässigkeitswachstumsmodellen für die Prognose ermessen zu können, sollten wir uns vor Augen führen, was wir von einem guten Prognoseverfahren erwarten. Nicht dass es uns so geht, wie dem Ingenieur, der zwar nicht genau weiß, was er misst, aber er tut es auf 10 Stellen genau. Und schließlich glaubt er, was er da liest.

Wissen ist *synthetisch* (auch: *empirisch*) insofern, als gewisse Annahmen eingehen, die nicht a priori gültig sind und die durch Erfahrung bestätigt sein müssen. Es ist *analytisch* in den Teilen, die allein auf logischen Schlussfolgerungen und Mathematik beruhen.

Den synthetischen (empirischen) Gehalt von Prognoseverfahren misst man am besten an den Kriterien für wissenschaftliche Theorien. Dazu gehören nach K. R. Popper (1982) die folgenden.

- *Falsifizierbarkeit:* Ein Prognoseverfahren muss einen Vorhersagewert haben und an der Erfahrung wenigstens prinzipiell scheitern können. Nicht so etwas wie: „Wenn der Hahn kräht auf dem Mist, ändert sich das Wetter oder es bleibt wie es ist."
- *Bewährtheit:* Ein (falsifizierbares) Prognoseverfahren, dessen Vorhersagewert sich in vielen unterschiedlichen Fällen praktisch erwiesen hat, halten wir für bewährt.
- *Objektivität:* Aussagen (Prognosen) müssen intersubjektiv nachprüfbar sein.
- *Einfachheit:* Ein Prognoseverfahren darf nicht von zu vielen anpassbaren Parametern abhängen, sonst entgeht es zu leicht der Widerlegung und hat einen zu geringen Vorhersagewert.

Ausfallratenaddition zur Zuverlässigkeitsprognose von Hardware. Genügt die Ausfallratenaddition (Abschnitt 3.5) den genannten Kriterien? Grundsätzlich schon, denn: Über das Ausfallverhalten der ausgelieferten Geräte im Betrieb werden üblicherweise Statistiken geführt. Sollten diese Daten signifikant von den prognostizierten abweichen, muss die Theorie verworfen werden. Also: Die Zuverlässigkeitstheorie ist grundsätzlich *falsifizierbar*.

Die Prognose ist eine Hypothese, die statistischen Tests unterzogen werden kann. Entgeht die Prognose der Widerlegung, kann die elementare Zuverlässigkeitstheorie als *bewährt* gelten. Die Prognose geschieht nach festen und allgemein anerkannten Regeln, ebenso die Überprüfung der Theorie. Also ist auch die *Objektivität* sichergestellt. Das Prognoseverfahren ist *einfach*. Einzig der Nachweis der Anwendbarkeit der elementaren Zuverlässigkeitstheorie und die Anpassung der Komponenten-Ausfallraten an die tatsächlich gegebenen Randbedingungen (beispielsweise Umgebungs- und Stressfaktoren) bereiten einige Schwierigkeiten.

11.8 Retrospektive statt Prognose

Ausfallraten von Bauelementen können gemessen werden: Die Bauelemente werden in vielen Exemplaren bei weitgehend gleichbleibenden Bedingungen nach einem bestimmten Muster hergestellt. Das macht die statistische Erfassung des Ausfallverhaltens überhaupt erst möglich und sinnvoll. Diese Daten machen den empirischen Gehalt und den Vorhersagewert der Zuverlässigkeitsmodelle für Hardware aus.

Mit den Ausfällen von Bauelementen haben wir also *Erfahrungen* - nicht aber mit den eingebauten Fehlern, um die es bei den Zuverlässigkeitswachstumsmodellen geht: Wir wissen nichts darüber, denn sonst hätten wir sie ja vermieden. Und dieses Unwissen hängt den Zuverlässigkeitswachstumsmodellen als Makel an.

Prognosen des Zuverlässigkeitswachstums auf Basis der Modelle setzen fehlerbehaftete Software und einen stabilen Software-Produktionsprozess voraus. Boshaft gesagt: Die Modelle sind umso zuverlässiger je unzuverlässiger die Software ist und je besser man sich auf diese Unzuverlässigkeit verlassen kann. Das gilt jedenfalls für ihre Qualität als Prognosewerkzeug.

Der Ingenieur aber will Software möglichst ohne Fehler konstruieren. Toleranz gegenüber selbstfabrizierten Fehlern gehört nicht zu den Tugenden des Ingenieurs. Und wenn der Software-Produktionsprozess fehlerhafte Software liefert, muss man die Regeln des Prozesses

radikal ändern. Und genau durch solche Änderungen werden die Zuverlässigkeitswachstumsmodelle als Prognoseinstrument unbrauchbar.

Bisher sind alle Versuche gescheitert, einen Zusammenhang zwischen den Merkmalen des Software-Herstellungsprozesses und den Zuverlässigkeitswachstumsmodellen herzustellen. Allgemein akzeptiert ist der Standpunkt, dass es nicht möglich ist, allein aus den Bedingungen, unter denen ein System entsteht, herauszufinden, welches der Modelle das passende für eine Zuverlässigkeitsprognose ist.

Brocklehurst und Littlewood drücken das im Buch von Lyu (1996) so aus: „Es gibt kein allgemein akzeptiertes Modell, dem zuzutrauen ist, dass es unter allen Umständen genaue Ergebnisse liefert; Anwender sollten gegenteiligen Versprechungen misstrauen. Schlimmer noch, wir können für eine bestimmte Datenquelle nicht a priori das Modell der Modelle herausfinden, das genaue Ergebnisse liefert, falls es ein solches überhaupt gibt; wir verstehen einfach nicht, welche Faktoren die Genauigkeit eines Modells beeinflussen." (S. 156).

Der letzte Schrei auf dem Gebiet der Zuverlässigkeitsmodellierung ist eine Strategie der *fortlaufenden Anpassung von Modellstruktur und Modellparameter* an das im Laufe der Testphase anfallende Datenmaterial. Damit wird der Zusammenhang zwischen den Fehlerursachen und den ermittelten Schätzwerten weiter vernebelt.

Das Lernen aus den Fehlern der Vergangenheit ist eine bewährte Methode der Fehlervermeidung. Das gilt für die Ingenieurwissenschaften wie in der Natur. Besonders fruchtbar ist eine genau Ursachenanalyse der Fehler. Zuverlässigkeitswachstumsmodelle geben da nichts her: „Die Modelle enthalten nicht genügend Information, die uns bei der Erklärung der Ursachen eines Problems helfen könnte; schlimmer noch, sie können uns nicht einmal sagen, ob es überhaupt ein Problem gibt. Das Einzige, was die Modelle tun, ist, einen Trend aufzuzeigen. Aber es bleibt dem Anwender der Zuverlässigkeitswachstumsmodelle überlassen, seine Bedeutung zu erfassen" (Miranda, 1998).

Also: Zuverlässigkeitswachstumsmodelle eignen sich weder besonders gut zur Zuverlässigkeitsprognose, noch sind sie geeignet, den Fehlerursachen beizukommen. Aber eins kann man mit Zuverlässigkeitswachstumsmodellen sehr wohl erreichen: Eine Beurteilung der Güte des Zuverlässigkeitsmanagements, also der Gesamtheit der Maßnahmen zur Fehlererkennung und Fehlerbeseitigung.

Wir lenken unseren Blick jetzt also auf die Beurteilung eines Software-Produktionsprozesses. Die Maßnahmen der Zuverlässigkeitserhöhung, insbesondere die Fehlerentdeckung und -beseitigung, mögen in der unter Beobachtung stehenden Phase in bestimmter und unveränderter Weise geregelt sein. Dieses Zuverlässigkeitsmanagement wird sich in einem Trend des Zuverlässigkeitswachstums niederschlagen. Und dieser Trend lässt sich mit den oben beschriebenen einfachen Zuverlässigkeitswachstumsmodellen ermitteln.

Die Modelle dienen also der Beurteilung unserer Anstrengungen, die in der *Vergangenheit* liegen: Hat die Regressionsgerade eine negative Steigung, spricht das gegen die ergriffenen Maßnahmen. Ist die Steigung positiv, haben die zuverlässigkeitserhöhenden Maßnahmen gegriffen - und je größer die Steigung, umso mehr. Auch das Duane-Modell lässt sich *retrospektiv* anwenden: Mit dem Parameter α passen wir die Kurve an die Daten an. Aus Tabelle 11.1 können wir dann das Urteil über die bisher ergriffenen zuverlässigkeitserhöhenden Maßnahmen ablesen. Die naive Zuverlässigkeitsschätzung lässt sich in derselben Art und Weise nutzen: Man interpretiert die Werte einfach als gleitende Mittelwertbildung und kann aus dem Verlauf dieses Maßes direkt auf Erfolg und Mißerfolg schließen.

12 Risiko

Entwickler, Konstrukteure, Inbetriebnahme- und Wartungsingenieure, Operateure, Marketingexperten und Manager stehen immer wieder vor Entscheidungen, deren Konsequenzen erst die Zukunft zeigen wird. Es geht bei diesen Entscheidungen beispielsweise um

- Lagerhaltungs- und Marktstrategien
- Produkteigenschaften (Schieberegler oder Drehknopf?)
- zu verwendende Materialien (recycelbar oder nicht?)
- Entwurfsalternativen und konstruktive Merkmale
- Fehlerbehebungsmaßnahmen bei ungewisser Diagnose
- Fahrtrouten eines Großschiffes (einer Gefahr ausweichen oder direkt durch?)
- Betriebsweisen eines Kraftwerks oder einer verfahrenstechnischen Anlage bei einer drohenden Katastrophe.

12.1 Fallsammlung von Fehlentscheidungen

Voreilige Markteinführung. Um die Stellung als Marktführer nicht zu verlieren, bringt das Management das neue Software-Produkt voreilig auf den Markt. Schließlich laufen die Kunden weg. Das Produkt hat noch zu viele Mängel.

Die „Vergoldete Lösung“. Die Suche nach der „vergoldeten Lösung“ hält die Entwicklungsabteilung lange auf. Das Produkt kommt viel zu spät auf den Markt. Der Konkurrent ist mit seinem - zugegeben etwas weniger komfortablen Produkt - bereits Marktführer geworden.

Tanker im Ärmelkanal. „Ein Frachter wollte Zeit sparen, fuhr einen falschen Fahrweg und kollidierte mit einem Öltanker, der explodierte, so dass noch 10 km entfernt in Folkestone Fensterscheiben zu Bruch gingen. Auch der Frachter ging auf Grund, aber da der Kanal ein relativ seichtes Gewässer ist, stellte er für die anderen Schiffe ein gefährliches Hindernis dar und wurde deshalb mit Warnbojen markiert [und nach weiteren Zwischenfällen durch zwei Feuerschiffe gesichert]... Zwei Wochen später ignorierte ein Tanker unbekannter Identität eine Sperre aus Leuchtraketen und Blinklichtern der beiden Feuerschiffe, durchpflügte eine Reihe von Leuchtbojen und schaffte es zur Überraschung aller, unbeschädigt durchzukommen und in der Nacht zu verschwinden... Mittlerweile waren an dieser Stelle 47 Menschen ums Leben gekommen. Nach Berichten der englischen Küstenbehörden hatten innerhalb von zwei Monaten 16 Schiffe die Warnzeichen ignoriert und ihren Weg durch das Unfallgebiet genommen - es war die schnellste Route“ (Perrow, 1987, S. 246-247).

Tschernobyl. In den bekannten Störfallablaufanalysen des Reaktorunfalls von Tschernobyl wird immer wieder darauf hingewiesen, dass der Reaktor für einen Test vorsätzlich außerhalb der Spezifikation und unter bewusster Verletzung der Sicherheitsvorschriften betrieben worden ist (Medwedjew, 1991, S. 38 ff.). Auslöser des Unfalls war die Entscheidung, einen Test durchzuführen, der außerordentlich riskante Bedienhandlungen verlangte.

Riskante Manöver nach dem Motto „Augen zu und durch“ gibt es nicht nur beim Betrieb von Anlagen und Systemen mit sehr hohem Gefährdungspotential. Auch aus dem Alltagsleben kennen wir sie: Bei Überholvorgängen im Straßenverkehr kommt es oft zu einer hohen Gefähr-

dung des eigenen Lebens und des Lebens anderer. Dem steht meist ein recht bescheidener Nutzen, ein geringfügiger Zeitgewinn, gegenüber.

Challenger. Am 28. Januar 1986 explodierte kurz nach dem Start die Raumfähre Challenger. Die siebenköpfige Besatzung kam dabei um. Als Ursache des Unglücks wurde ein Leck in einer der beiden Feststoffraketen ausgemacht. Das Leck trat an einer Verbindungsstelle zwischen zwei Raketensegmenten auf. Diese Verbindungen waren lange vor dem Unglücksflug als kritisch erkannt worden. Die Startfreigabe war eine Fehlentscheidung nicht eines Einzelnen, sondern einer ganzen Organisation.

Eschede. Am 3. Juni 1998 um 10.59 Uhr verunglückte der ICE Wilhelm Conrad Röntgen in Eschede bei Celle; 101 Tote und 119 größtenteils schwer Verletzte sind die Bilanz dieser Zugkatastrophe - eine der schwersten der deutschen Nachkriegsgeschichte. Hauptursache war ein zerbrochener Radreifen. Der Einsatz der Radreifen dieser Bauart und auch die Rückstellung der Ersetzung des betroffenen Exemplars stellten sich im Nachhinein als Fehlentscheidungen heraus. Die Entscheidungen wurden auf unzureichender Wissensbasis getroffen, denn es lag keine ausreichende Untersuchung der Ermüdungsfestigkeit der Radreifen vor. An den Fehlentscheidungen wirkten mehrere Stellen innerhalb der Organisation der Deutschen Bahn mit (Hörstel/Ritzau, 2000).

12.2 Entscheidung bei Risiko

Wie alle unsere Schwächen so ist auch die Lust am Risiko nur die Kehrseite einer an sich nützlichen Verhaltensweise: „Exploration ist die Triebhandlung des Sicherheitstriebes, also die mit Anstrengung verbundene Umwandlung der Unsicherheit in Sicherheit“ (von Cube, 1995, S. 76). Exploration ist stets mit Risiken verbunden. Wir müssen Risiken eingehen, um uns die Welt vertraut zu machen, und um noch größere Risiken zu vermeiden. Gefahr droht also dann, wenn wir keine Risiken eingehen wollen, aber auch dann, wenn wir Risiken nicht richtig einschätzen.

Aber: Wie sind Risiken zu bewerten? Gibt es überhaupt ein Maß dafür? Wie sollte man es definieren?

Wir beginnen mit einer Feststellung: *Risiko* quantifiziert die Angst vor einem möglichen Schaden. Und diese Angst wächst nicht nur mit der Schwere des befürchteten Schadens, sondern auch mit der Wahrscheinlichkeit, mit der dieses Ereignis eintreten kann. Als vorläufiges Maß für das Risiko R nehmen wir das Produkt aus Schadenswahrscheinlichkeit P und Schadenshöhe S, also

$$R = P \cdot S$$

Risikomanagement besteht darin,

1. Alternativen zur Erzielung eines Nutzens oder zur Vermeidung eines Nachteils aufzuzeigen,
2. die mit den Alternativen verbundenen Risiken zu quantifizieren, und
3. diejenige Alternative auszuwählen, mit der das geringste Risiko verbunden ist.

Machen wir uns das Wesentliche des Risikomanagements an einer alltäglichen und wenig furchteinflößenden Situation klar.

Horst Hemmerling hat für den Samstagabend Freunde zu sich eingeladen. Als Höhepunkt des Festes hat er eine kleine Diaschau über den gemeinsam verbrachten Urlaub vorgesehen. Alle freuen sich darauf. Am Samstagvormittag fällt Horst ein, dass er für den Diaprojektor keine Ersatzbirne mehr hat. Er denkt kurz nach: Gehe ich jetzt noch einmal in die Stadt und hole eine

Ersatzbirne, kann ich eine andere Besorgung nicht mehr machen. Hole ich sie nicht, könnte die Diavorführung platzen. Um die Situationen vergleichen zu können, drückt er die möglichen Nachteile in Euro aus. Er fragt sich, was er maximal spendieren würde, um den jeweiligen Nachteil oder Schaden zu vermeiden. Er kommt auf folgende Werte:

- Diaschau geplatzt: 500 Euro.
- Wichtige Besorgung nicht gemacht: 50 Euro.

Jetzt fragt er sich, wie wahrscheinlich das Versagen der Birne ist. Gemessen am Alter der Birne rechnet er damit, dass sie mit der Wahrscheinlichkeit von 5% versagen wird. Er stellt seine Entscheidungssituation in Tabellenform dar (Tabelle 12.1).

Für jeden möglichen Fall multipliziert er die Wahrscheinlichkeit des Eintretens des Falles mit den dabei entstehenden Kosten und erhält damit das (Teil-)Risiko dieses Falles. Für jede der Entscheidungsalternativen (ja/nein) bildet er die Summe dieser Teilrisiken und erhält so das Risiko der jeweiligen Entscheidungsalternative. Das Risiko der Ja-Entscheidung (Kauf) ist deterministisch - und damit im eigentlichen Sinne des Wortes kein Risiko. Es beträgt 50 Euro. Das Risiko der Nein-Entscheidung (Verzicht auf den Kauf) beträgt nur 25 Euro. Für Horst ist die Sachen nun klar: er entschließt sich, auf die Ersatzbirne zu verzichten.

Bereits dieses einfache Beispiel lässt grundlegende Probleme des Risikomanagements hervortreten: Das Risiko erfasst bestimmte Befindlichkeiten des Entscheiders nicht. Wäre Horst ein etwas ängstlicher (fachmännisch: risikoaverser) Typ, würde er vielleicht trotz seiner Überlegungen eine Ersatzbirne kaufen gehen.

Tabelle 12.1 Eine einfache Entscheidungssituation in Tabellenform

Ersatzbirne kaufen?			*Wahrscheinlichkeit*	*Kosten (in Euro)*	*Risiko (in Euro)*
	ja				
			100%	50	50
				Summe:	50
	nein				
		Birne bleibt heil	95%	0	0
		Birne geht kaputt	5%	500	25
				Summe:	25

Zur Bestimmung der Wahrscheinlichkeiten braucht man die Erfahrungen mit vergleichbaren Fällen aus der Vergangenheit. Nicht immer liegen diese vor. Besonders schwierig wird es bei den ganz kleinen Wahrscheinlichkeiten für Ereignisse mit katastrophalen Auswirkungen: Hier will man möglichst keine Erfahrungen sammeln. Das ist ein Grundproblem, mit dem alle Techniken mit hohem Gefährdungspotential zu kämpfen haben: Schnellbahn, Überschallverkehrsflug, Kernkraft, Chemie.

Das Verrechnen von nicht-monetären und ideellen Werten auf einer einheitlichen Skala gehört zu den besonders schweren Aufgaben der Risikoanalyse (Fritzsche, 1986, S. 52 ff.). Heikel ist das „Verrechnen“ von möglichen Todesfällen und wirtschaftlichen Vorteilen (siehe „riskante Manöver“).

12.3 Das objektive Risiko

Nehmen wir einmal an, dass die mit allen Alternativen einhergehenden Kosten oder Schäden für jeden einsichtig dargelegt und mit Maßzahlen belegt werden können. In diesem Fall be-

zeichnet man das errechnete Risiko als *objektives Risiko*. Die Beifügung „objektiv“ ist keine positive Hervorhebung. Sie besagt lediglich etwas über die eingeschränkte Anwendbarkeit dieses Begriffes: Nur dort, wo objektive (d.h.: intersubjektiv nachprüfbare) Daten vorliegen und die besondere Lage der Betroffenen und der Entscheider keine Rolle spielen, ist das objektive Risiko ein tragfähiger Begriff. Er meint den *Schadenserwartungswert*.

Mit einer Entscheidung, einer technischen Anlage, einem Manöver oder dergleichen möge ein zufälliger Schaden verbunden sein. Der Schaden lässt sich also durch eine Zufallsvariable X beschreiben, die wir uns hier einmal als diskret vorstellen wollen. Mit gewissen Schadensfällen i seien Schäden der Höhe x_i verbunden, die mit den Wahrscheinlichkeiten p_i eintreten. Das objektive Risiko ist dann definiert durch

$$R = E[X] = \Sigma_i \, p_i \, x_i.$$

Ein Entscheider hat zwei oder mehr Alternativen zur Auswahl. Jede der Alternativen ist mit einem (zufälligen) Schaden verbunden. Auch entgangener Nutzen wird zu den Schäden gezählt. Wir beschränken uns auf zwei Alternativen und nennen diese A und B. Die Alternative A habe den zufälligen Schaden X, die Alternative B den zufälligen Schaden Y zur Folge. Die mit den Entscheidungen A und B verbundenen *objektiven Risiken* sind $E[X]$ und $E[Y]$. Durch sie wird eine *Präferenzordnung* der Alternativen festgelegt: A wird gegenüber B genau dann bevorzugt, wenn $E[X] < E[Y]$ ist.

Der Begriff des objektiven Risikos kommt aus dem Versicherungswesen (Bernstein, 1996) und er bildet die Grundlage der *technischen Risikoanalysen* (Risk Assessment).

12.4 Der Entscheidungsbaum

Für kleinere Entscheidungsprobleme ist die Tabelle ein geeignetes formales und rechnerisches Hilfsmittel. Die Baumstruktur, die einer *Entscheidung bei Risiko* zugrunde liegt, kommt aber am ehesten in einer Verallgemeinerung des Ereignisbaums zum Ausdruck: Der *Entscheidungs-Ereignisbaum* (kurz: Entscheidungsbaum) ist ein Baum mit zwei Sorten von Knoten, eine für die mehr oder weniger zufälligen Ereignisse und eine für die Entscheidungen (Hammond/Keeney/Raiffa, 1999).

Wie sich eine Entscheidungssituation mittels Entscheidungs-Ereignisbaum beschreiben und analysieren lässt, soll das folgende Beispiel einer Störungsdiagnose und -beherrschung in einem Kraftwerk zeigen.

Der Operateur eines Kohlekraftwerks steht vor folgender Entscheidungssituation: Er beobachtet, dass das Abflussventil des ersten Hochdruckvorwärmers sich zu öffnen beginnt, das des zweiten beginnt sich zu schließen. Als Ursachen für dieses abnorme Verhalten kommen zu diesem Zeitpunkt zwei Hypothesen in Betracht, nämlich

H_0: Es gibt ein kleines Leck in einer Speisewasserleitung des ersten Hochdruckvorwärmers. Die Wahrscheinlichkeit dafür ist $P(H_0) = 5\%$.

H_1: Es liegt irgend eine andere - eher harmlose - Ursache vor. Die Wahrscheinlichkeit dieser Hypothese ist gleich $P(H_1) = P(\neg H_0) = 95\%$.

Der Operateur sieht die folgenden Entscheidungsalternativen:

A0 Sofortige Abhilfemaßnahme gegen Leckage.

A1 Durchführung einer Diagnose auf Leckage vor der endgültigen Entscheidung.

A2 Abhilfemaßnahme gegen Leckage nach einem positiven Untersuchungsergebnis.

A3 Keine Abhilfemaßnahme auch bei positivem Ergebnis der Diagnose.

Das Ereignis eines positiven Diagnoseergebnisses wird mit O bezeichnet. Im Falle eines negativen Ergebnisses ($\neg O$) wird nichts besonderes unternommen.

Im Falle eines Lecks ergeben sich folgende Kosten: Sofort sind es 5 Einheiten, die Verzögerung aufgrund der Diagnose kostet weitere 5 Einheiten; und falls auch dann nichts unternommen wird, kostet das Leck zusätzlich 90 - also insgesamt 100 - Einheiten. Die Abhilfemaßnahme gegen die Leckage kostet 5 Einheiten. Der Erwartungswert der Kosten im Falle einer anderen Ursache beträgt 2 Einheiten. Tabelle 12.2 zeigt eine Übersicht über die Kosten, die mit den verschiedenen Alternativen und Aktionen verbunden sind. Die Kosten sind von der Störungsursache abhängig.

Tabelle 12.2: Kosten der Entscheidungsalternativen

Aktionen	H_0	H_1
A0	10	7
A1	100	2
A1, A2	15	7
A1, A3	100	2

Durch die Diagnose möge ein eventuelles Leck mit einer Irrtumswahrscheinlichkeit von 0.2 erkennbar sein: Sowohl die Falsch-positiv-Wahrscheinlichkeit $P(O|H_1)$ als auch die Falsch-negativ-Wahrscheinlichkeit $P(\neg O|H_0)$ betragen je 20%.

Die Wahrscheinlichkeit eines positiven Diagnoseergebnisses ist gleich

$$P(O) = P(O|H_0)P(H_0) + P(O|H_1)P(H_1) = 0.8*5\% + 0.2*95\% = 23\%$$

Die bedingten Hypothesenwahrscheinlichkeiten in Abhängigkeit vom Diagnoseergebnis sind

$$P(H_0|O) = P(O|H_0)P(H_0)/P(O) \approx 17\%,$$

$$P(H_1|O) \approx 83\%,$$

$$P(H_0|\neg O) = P(\neg O|H_0)P(H_0)/P(\neg O) \approx 1\% \text{ und}$$

$$P(H_1|\neg O) \approx 99\%.$$

Bild 12.1 zeigt den *Entscheidungs-Ereignisbaum* der Diagnoseaufgabe. Rechtecke stehen für *Entscheidungszustände* und Kreise für die Ereignisse. Direkt über den Pfaden stehen die jeweiligen Entscheidungsalternativen bzw. die bedingten Wahrscheinlichkeiten für das Durchlaufen des Pfades. Neben den Blättern (Endknoten) des Baums stehen die Kosten für den gesamten Pfad ab Wurzel.

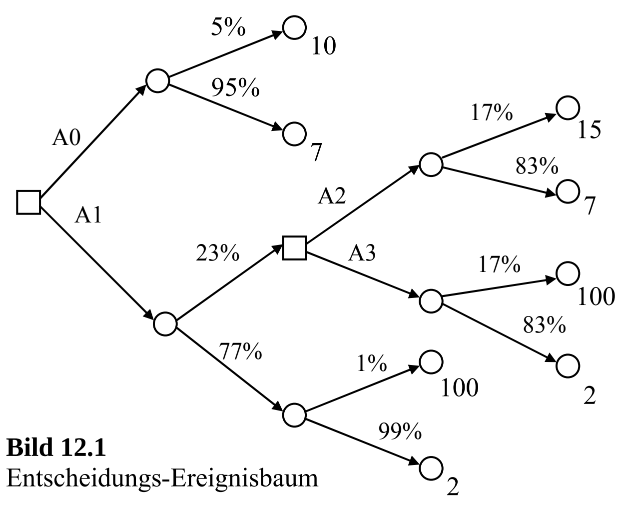


Bild 12.1
Entscheidungs-Ereignisbaum

Rekursiv lassen sich allen Knoten des Baumes optimale - also minimale - Risikowerte zuweisen: Bei den Teilbäumen, die reine Ereignisbäume sind, geht das nach den Pfadregeln (Kapitel 8). Bei den Entscheidungsknoten wird der Optimalwert durch die am wenigsten riskante Entscheidung bestimmt: Es wird der Nachfolgeknoten mit dem niedrigsten Risikowert ausgewählt, und dieser Wert wird für den Entscheidungsknoten übernommen.

Die Ermittlung der Optimalwerte und der optimalen Entscheidungen geschieht in unserem Beispiel demnach so: Zunächst werden die zu den Entscheidungen A0, A2 und A3 gehörenden

Ereignisbäume ausgewertet. Es ergeben sich die Risiken 7.15, 8.4 und 18.7. Nun lässt sich auch das Risiko der Entscheidung A1 ermitteln. Der Wert ist gleich 4.22.

Das heißt: Der Operateur sollte sich für eine Diagnose entscheiden, da A1 ein geringeres Risiko darstellt als A0. Falls das Diagnoseergebnis positiv ist, sind die Schritte zur Bekämpfung eines Lecks einzuleiten, denn A2 beinhaltet ein geringeres Risiko als A3.

Der Entscheidungs-Ereignisbaum sagt, was der Bediener tun sollte; jede Abweichung davon wird als Bedienfehler gewertet. Der Entscheidungs-Ereignisbaum spielt also die Rolle eines *normativen Modells des Bedienerverhaltens*.

Das normative Modell sagt nicht, wie sich der Bediener tatsächlich verhält. Vielmehr stellt es die Messlatte dar, an der das tatsächliche Bedienerverhalten zu messen ist. Der Nutzen des normativen Modells des Bedienerverhaltens liegt darin, dass es überhaupt erst ermöglicht, über *Bedienfehler*, also Abweichungen von der Norm, zu reden.

Anders als die Entscheidungen im Rahmen von Planungsprozessen kommen die Entscheidungen eines Operateurs im Allgemeinen unter erheblichem Zeitdruck zu Stande. Normative Modelle des Bedienerverhaltens können nicht erst dann entwickelt werden, wenn die Probleme schon anliegen. Eine Rolle können sie beispielsweise bei der Planung von psychologischen Experimenten zur Erfassung des Bedienerverhaltens spielen (Grams, 2000).

12.5 Rationale Entscheidungen bei subjektivem Risiko

Der bisher behandelte Risikobegriff enthält innere Widersprüche. Wir entscheiden tatsächlich nicht immer nach dem objektiven Risiko. Ein Beispiel: Der Chef einer kleinen eher schlecht gehenden Firma möge die Chance haben, mit einer noch unzureichend ausgetesteten Software ordentliche Gewinne zu machen. Die Gefahr, dass Regressforderungen ihn in den Konkurs treiben, veranschlagt er mit 1%. Ein risikofreudiger Typ wird die Sache auch dann machen, wenn das objektive Risiko dagegen spricht. Und es wäre eine durchaus *rationale Entscheidung!*

Dieses Beispiel ist nicht aus der Luft gegriffen. Psychologische Experimente belegen, dass die meisten Menschen risikofreudig sind, wenn es um Verluste geht. Hier das Ergebnis eines solchen Experiments: Ein Verlust von 4000 Israelischen Pfund, der mit 80-prozentiger Wahrscheinlichkeit eintritt, wird von 92% der Versuchspersonen einem sicheren Verlust von 3000 Pfund vorgezogen. Und das, obwohl das Risiko 3200 anstelle von 3000 Pfund beträgt (Kahneman/Tversky, 1979).

Anhand des *St. Petersburger Spiels* wurde dem Mathematiker Daniel Bernoulli 1738 klar, dass das objektive Risiko als Richtschnur für Präferenzordnungen nicht ausreicht (Székely, 1990, S. 34 ff.). Er kam zu einer Lösung, die von der Spieltheorie und von der betriebswirtschaftlichen Entscheidungslehre aufgegriffen worden ist: Bernoulli führte die Nutzenfunktion ein, die mit u (Utility) bezeichnet wird. Wenn z ein objektiver Nutzen ist, dann wird mit $u(z)$ der subjektiv empfundene Nutzen bezeichnet.

Hier wird, wie es in der technischen Risikoabschätzung üblich ist, vom Schaden und nicht vom Nutzen ausgegangen. Das Problem der *Entscheidung bei Risiko* wurde dementsprechend umformuliert. Die Rolle der Nutzenfunktion wird nun von der *subjektiven Schadensfunktion s* übernommen.

Das *subjektive Risiko* $R_{\text{subjektiv}}$ einer Entscheidung A, die mit dem zufälligen (objektiven) Schaden X verbunden ist, ist gleich dem Erwartungswert des subjektiven Schadens:

$$R_{\text{subjektiv}} = E[s(X)] = \Sigma_i\, p_i\, s(x_i).$$

Die *Präferenzordnung* der Alternativen wird nun durch den Erwartungswert des subjektiven Schadens festgelegt: Alternative A wird gegenüber B genau dann bevorzugt, wenn $E[s(X)]<E[s(Y)]$ ist, wobei Y der mit der Alternative B verbundene Schaden ist.

Allgemeine Überlegungen legen nahe, für die subjektive Schadensfunktion $s(x)$ eines risikofreudigen Entscheiders einen Ansatz zu machen, der für kleine Werte von x in etwa linear ist, und der für große Werte in einen logarithmischen Verlauf übergeht. Der Übergangsbereich wird durch einen Ankerwert x_0 festgelegt. Mit einer Konstanten c ist $s(x) = c \cdot \ln(1+x/x_0)$ ein brauchbarer Ansatz. Es handelt sich bei $s(x)$ um eine konkave Funktion, also um eine Funktion, deren Steigung mit wachsendem x abnimmt. Der Ankerwert x_0 liegt in der Übergangszone zwischen den Gültigkeitsbereichen des linearen und des logarithmischen Schadensgesetzes.

Da die Präferenzordnung durch die Konstante c nicht beeinflusst wird, kann sie beliebig gewählt werden. Sinnvoll ist es, sie so zu wählen, dass der größte in Frage kommende x-Wert einem subjektiven Schaden von 100% entspricht, Bild 12.2 (gewählter Ankerwert: 5000 Einheiten).

Ist die Schadensfunktion s streng konkav, dann gilt für jeden nichtdeterministischen Schaden X die *Ungleichung der Risikoakzeptanz:* $E[s(X)] < s(E[X])$. Sie besagt, dass bei gleichem objektiven Risiko der sicher eintretende Schaden als bedrohlicher angesehen wird als ein zufälliger Schaden.

Die Ungleichung der Risikoakzeptanz ist Ausdruck eines allgemeinen psychologischen Prinzips, nämlich der gut bestätigten *Tendenz zur Überbewertung der Gewissheit* (Overweighting of Certainty, Kahneman/Tversky, 1979).

Das *Sicherheitsäquivalent* eines zufälligen Schadens X bei gegebener subjektiver Schadensfunktion ist derjenige mit Gewissheit eintretende Schadenswert, der dem zufälligen Schaden gleichwertig ist. Bei Risikoakzeptanz ist das Sicherheitsäquivalent niedriger als das objektive Risiko. Für das Sicherheitsäquivalent x hat man die Formeln $s(x) = E[s(X)]$ oder

$$x = s^{-1}(E[s(X)]).$$

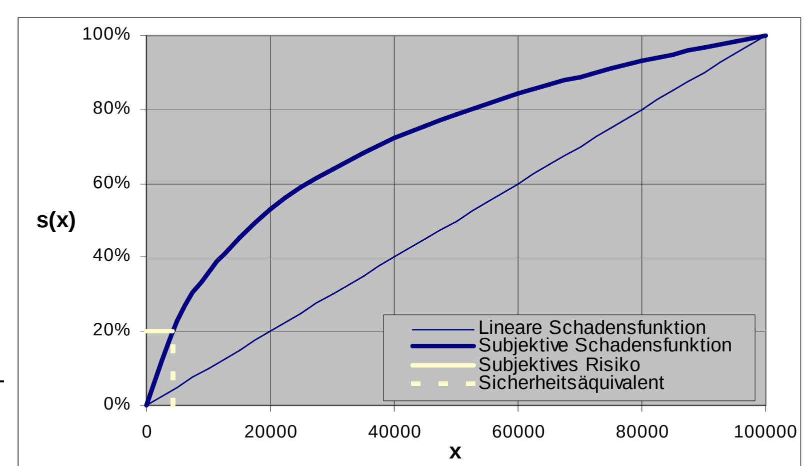


Bild 12.2 Konkave subjektive Schadensfunktion und Sicherheitsäquivalent

Beispiel: Wir legen die subjektive Schadensfunktion des Bildes 12.2 zugrunde und betrachten einen zufälligen Schaden X, bei dem mit der Wahrscheinlichkeit von 20% ein Schaden von 100 000 Einheiten entsteht. Mit der Wahrscheinlichkeit von 80% ist der Schaden gleich null. Das subjektive Risiko ergibt sich zu 20%, ausführlich: $E[s(X)] = 20\% \cdot s(100\,000) + 80\% \cdot s(0) =$

20%·100% + 80%·0% = 20%. Das Sicherheitsäquivalent ist derjenige feste Schadenswert x, dem dasselbe subjektive Risiko zugemessen wird. Da bei festem Schadenswert x dessen subjektives Risiko genau gleich dem subjektiven Schadenswert $s(x)$ ist, findet man den Wert x so: Man geht vom Ordinatenwert von 20% nach rechts, bis man auf die subjektive Schadensfunktion trifft, und von dort geht es nach unten zur Abszisse. Dort liest man den Wert x ab. Das Sicherheitsäquivalent liegt unter 5 000 Einheiten. Es ist deutlich geringer als das objektive Risiko mit einem Wert von 20 000 Einheiten.

Hinweise zur psychologischen Seite der Risikowahrnehmung bieten Fritzsche (1986) und Perrow (1987). Wichtige Ergebnisse zur Psychologie der Entscheidungspräferenzen haben Kahneman und Tversky beigesteuert. Die mathematischen Grundlagen der Entscheidungstheorie findet der Leser beispielsweise in den Büchern von Bamberg/Coenenberg (1989) und Carnap/Stegmüller (1959, S. 124 ff.).

12.6 Risikoakzeptanz

Bei technischen Risikoanalysen ist immer wieder von *Risikoaversion* die Rede (Fritzsche, 1986, S. 164-168). Risikoaversion entspricht einer konvexen subjektiven Schadensfunktion. Das scheint den allgemeinen Beobachtungen des vorigen Abschnitts zu widersprechen.

Unterschiede in der Risikowahrnehmung

Dieser Widerspruch löst sich auf. Wir müssen bedenken, dass wir es mit zwei verschiedenen Sachlagen zu tun haben:

1. *Entscheidung bei Risiko:* Bei der Risikobewertung geht es um die Möglichkeit *eines* Schadensfalls in dem *einen* System. Die als gering angesetzte Schadenswahrscheinlichkeit lässt den Unfall als „fast ausgeschlossen" erscheinen. Die Risikoschätzung wird von einem Experten durchgeführt, der die von ihm bediente Technik kennt. Die Risiken sind ihm *vertraut*. Der Experte setzt sich *freiwillig* der Gefahr aus und er kann das Risiko *beeinflussen*.
2. *Risikoabschätzung technischer Anlagen:* Hier steht das Kollektivrisiko im Zentrum der Überlegungen, und die Risikobewertung trägt der Tatsache Rechnung, dass es viele gleichartige Anlagen gibt. Der Schadensfall kann nicht mehr als „fast ausgeschlossen" angesehen werden, selbst wenn er in einer konkreten Anlage als sehr unwahrscheinlich erscheint. Die Risikobeurteilung muss die Wertvorstellungen von Unbeteiligten und Laien einbeziehen. Der Beurteilungsmaßstab ist mit geprägt von der Risikowahrnehmung derjenigen, die die Technik nicht richtig kennen und die ihr eventuell sogar *misstrauen*. Die meisten Betroffenen sind der Gefahr *unfreiwillig* ausgesetzt, und sie können das Risiko *nicht beeinflussen*.

Nun ist aber bekannt, dass die *Risikowahrnehmung* wesentlich durch

- die Bekanntheit einer Gefahr,
- die Freiwilligkeit im Eingehen eines Risikos und
- die Beeinflussbarkeit des Risikos

bestimmt wird. Die Ergebnisse empirischer Forschung legen tatsächlich nahe, im Fall 1 (Entscheidung bei Risiko) eher von Risikoakzeptanz und im Fall 2 (Risikoabschätzung technischer Anlagen) eher von Risikoaversion auszugehen (Fritzsche, 1986, S. 123-149).

Mit Risikofragen gehen die verschiedenen Akteure und Institutionen (Unternehmen, Öffentlichkeit, Politik, Wissenschaft und Technik) ganz verschieden um. Wie wir sehen, können sie

auf ganz rationale Weise zu durchaus unterschiedlichen Einschätzungen ein und derselben Sache kommen. Die damit einhergehenden Kommunikationsprobleme werden im Buch von Renn/Zwick (1997) umfassend dargestellt.

Welches Risiko ist vertretbar?

Der Staat hat durch Gesetz, Verordnungen und Vorschriften Schutzziele festzulegen unter der Maßgabe, dass von der Technik keine Gefahr für die Bevölkerung ausgeht. Welche Risiken darf der Gesetzgeber der Bevölkerung zumuten? Was dem einen nach reiflicher Überlegung als noch vertretbar erscheint, ist für den anderen (aus ebenso rationalen Gründen) bereits jenseits alles Verantwortbaren.

Unter *Gefahr* wird eine Sachlage verstanden, „bei der das Risiko größer als das Grenzrisiko ist, wobei unter *Grenzrisiko* das größte noch vertretbare Risiko verstanden wird“ (DIN VDE 31 000, Teil 2; VDI/VDE 3542, Blatt 2). *Sicherheit* ist komplementär zur Gefahr. Sie ist gegeben, wenn das Risiko nicht größer als das vertretbare Risiko ist. Die Frage der Sicherheit spitzt sich folglich auf die Frage nach dem noch *vertretbaren* - oder akzeptablen - *Risiko* zu. Es gibt eine ganze Reihe von „Vorschlägen für ein akzeptables Risiko“ Fritzsche (1986, S. 67 ff.).

Beispielsweise beantwortet Starr (1969) die Frage „Wie sicher ist sicher genug?“ unter Bezugnahme auf die Alltagsrisiken und die Risiken, die der Betroffene freiwillig übernimmt. Ihm ist klar, dass im Falle freiwilliger Aktivitäten das Individuum sein eigenes Wertesystem für die Risikoschätzung verwendet: „Wie zu erwarten, lassen wir andere uns ungern das antun, was wir ohne Zögern uns selbst zumuten... Es gibt Hinweise, dass die Öffentlichkeit 'freiwillige' Risiken akzeptiert, die ungefähr 1000 mal so groß sind wie die 'unfreiwilligen' Risiken.“ Seine Schätzungen des akzeptierten Risikos gehen von 1 Todesfall pro 10^9 Personenstunden des Ausgesetztseins bei der Kerntechnik (unfreiwillig) bis zu 1 Todesfall pro 10^6 Personenstunden des Ausgesetztseins im Falle des Autoverkehrs (freiwillig).

Der Ansatz einer allgemeinen Risikogrenze, wie er von Starr vertreten wird und wie er im aktuellen Normenwerk zum Ausdruck kommt, ist umstritten. Einem gegensätzlichen Standpunkt zufolge sollte kein Risiko, das noch leicht reduziert werden könnte, als akzeptabel bezeichnet werden.

Dieser Standpunkt kommt im *ALARA-Prinzip* der Risikofestlegung zum Ausdruck, also im Grundsatz „As Low As Reasonably Achievable“: Jedes neue Risiko muss so weit reduziert werden, wie der damit verbundene Aufwand noch als vernünftig erscheint.

Offensichtlich hängt die Risikoakzeptanz von Bedürfnissen, Interessen, Wertvorstellungen, Besorgnissen, Ängsten und Wünschen der Betroffenen und der Entscheider ab. Die Frage nach der Sicherheit technischer Produkte berührt den Bereich der Moralphilosophie. Es ist eine Frage der Ethik, wie viel Risiko man anderen zumuten darf.

Die Moral des Risikos

Die *goldene Regel* („Alles nun, was ihr wollt, dass euch die Leute tun sollen, das tut ihnen auch!“, Matthäus 7, 12) ist angesichts der Subjektivität der Risikowahrnehmung keine hilfreiche Verhaltensmaxime.

Niklas Luhmann (1993) spricht es deutlich aus: „Was die eigenen Entscheidungen angeht, ist man oft extrem risikobereit: Man fährt Auto oder sogar Motorrad, man klettert auf Berge, man heiratet. Bei Gefahren, die einem von anderer Seite zugemutet werden, ist man dagegen hochempfindlich ... Übersetzt in das Thema Moral heißt dieser 'double standard': Es gibt keine sinnvolle Anwendung für die Maxime der Reziprozität ... Wenn noch gälte: 'Liebe Deinen Nächsten wie dich selbst', könnte dieser sich auf allerhand gefasst machen“.

Mit der goldenen Regel verlieren wir den letzten Rest an festem Grund: „Die Unsicherheit des reflektierenden Menschen ist zwangsläufig total“ (Felix von Cube).

Diese Erkenntnis ist ernüchternd. Aber sie passt ins Bild, das die Biologen und Verhaltensforscher von der Entstehung jeglicher Moral entwerfen. Es ist das Zusammenspiel von Angeborenem, kulturell Überliefertem und vernünftiger Voraussicht, das die Moralprinzipien hervorbringt und wandelt. Grundlegend sind die Mechanismen der Evolution - und diese beruhen nun einmal auf dem Prinzip von Versuch und Fehlerbeseitigung. Unsere Fähigkeit, die Zukunft in unserer Vorstellung zu simulieren, die Werkzeuge, Techniken und Methoden, die wir zur Verstärkung dieser Fähigkeit erdacht und gebaut haben, können uns helfen, die schmerzhaften Folgen dieses Urprinzips so gering wie irgend möglich zu halten.

Insgesamt erspart uns diese Sicht der Dinge die Suche nach übergeordneten und absoluten Prinzipien, denen die Ingenieursarbeit unterliegt. Eine allgemeingültige Antwort auf die Frage „Wie sicher ist sicher genug?“ kann es nicht geben. Die Sicht der Welt, die Einschätzung von Risiken und deren Erträglichkeit sind *soziale Konstruktionen*. Höchstrichterliche Begründungen dafür gibt es nicht, genauso wenig Widerlegungen. Das Einzige was man sagen kann, ist, dass sich die Konstruktionen im Wettbewerb der Ideen und ihrer Befürworter behaupten, oder eben nicht. Das heißt andererseits aber auch, dass wir die Verantwortung für unsere Regelwerke, Spezifikationen, Techniken und deren Auswirkungen nicht einer höheren Instanz übergeben können.

Risiko und Kultur

Die Entscheidung bei Risiko dient der „Vergegenwärtigung der Zukunft“. Ihr sind teils grundsätzliche und teils situationsabhängige Grenzen gesetzt:

1. Es kann schief gehen, auch wenn man richtig - und auf der Basis perfekter Informationen und bei vollem Konsens über die Schwere der Entscheidungskonsequenzen - entschieden hat.
2. Möglicherweise gibt es keinen Konsens über die Zuordnung von Kosten zu den Entscheidungskonsequenzen.
3. Es fehlen womöglich noch glaubwürdige Informationen und Daten bezüglich Art und Wahrscheinlichkeit der Entscheidungskonsequenzen.
4. Oft ist die Zeit für die Entscheidungsfindung zu knapp.

Punkt eins trifft den Kern aller Wahrscheinlichkeitsaussagen. Die durch Punkt zwei angesprochenen Probleme sind ebenfalls grundsätzlicher Natur. Sie wurden im letzten Abschnitt kurz angesprochen. Wie sieht es mit den Punkten drei und vier aus? Auch hier tut sich ein Einfallstor für Zweifel auf.

Der Mangel an glaubwürdigen Informationen lässt sich - in Grenzen vielleicht - durch Wissens- und Erfahrungserwerb beheben. Wenn Punkt vier nicht wäre! Das Produkt muss nun mal rechtzeitig auf den Markt. Bei schrillenden Alarmglocken ist an die Ausarbeitung eines normativen Entscheidungsmodells nicht zu denken. Das Challenger-Unglück und die Katastrophe von Eschede werfen ein grelles Licht auf die Tatsache, dass Start-, Konstruktions-, Diagnose- und Einsatzentscheidungen auf der Basis unvollkommener Information getroffen werden.

Gotthard Bechmann (1993) schreibt: „Jede Untersuchung braucht Zeit... Gleichwohl läuft die Entwicklung der Technik und ihre Integration in das Sozialgefüge der Gesellschaft weiter und wirft neue Probleme auf. Während dieser Zeit ändern sich sowohl die Fragestellungen als auch die Daten. Will man trotzdem zum Abschluss des Forschungs- und Entscheidungsprozesses kommen, muss man ab einem bestimmten Zeitpunkt neue Daten ignorieren“. Wenn alles im

Fluss ist, und wenn mit jeder Erkenntnis auch die Ungewissheit steigt, bleibt für Entscheidungen oft nur die Flucht in die Fiktion.

Und damit wird die Sphäre des Rationalen verlassen. Die Perspektive wird durch Gefühle, Überzeugungen, Gruppenzugehörigkeit, Glaubwürdigkeit der Informanten, gegenseitiges Vertrauen bestimmt. Ob jemand eine Entscheidung als richtig oder falsch ansieht, hängt vom gesellschaftlichen Umfeld, der Organisation und deren Kultur ab.

Mary Douglas und Aaron Wildavsky (1982) zeichnen in ihrem Buch „Risk and Culture" ein holzschnittartiges Bild der gesellschaftlichen Lage („The Center is Complacent - The Border is Alarmed"): Unter dem gesellschaftlichen „Zentrum" verstehen sie die hierarchisch oder marktwirtschaftlich organisierten herrschenden Institutionen. Und der gesellschaftliche „Rand" wird durch sektenartige Gruppierungen - etwa die Gruppe der „Fundamentalisten" - bestimmt.

Die Unentschiedenheit der Risikofrage lässt den Vertretern des gesellschaftlichen „Zentrums" Raum, das Risiko einer Technik zum „Restrisiko" herunterzureden und sich durch die Forderung nach wissenschaftlich zwingenden Nachweisen für vermutete Gefahren der Debatte zu entziehen.

Der gesellschaftliche „Rand" pflegt demgegenüber einen eher kompromisslosen Umgang mit dem Risiko nach der Devise „Sicherheit gibt es nie genug". „Angstkommunikation" (Niklas Luhman) alarmiert die Gesellschaft und sorgt gleichzeitig für den Gruppenzusammenhalt (Bechmann, 1993).

Risiko ist so gesehen ein soziales Konstrukt mit verschiedenen Ausprägungen. Die rivalisierenden Perspektiven wirken - durch Filterung von Fakten zur Stützung vorgefasster Meinungen - verstärkend auf sich zurück. Es kommt zur Ideologisierung und Polarisierung und zum Entstehen von miteinander unvereinbaren „Wagnis-" und „Sicherheitskulturen" (Renn/Zwick, 1997).

Trotz dieser eher pessimistisch stimmenden Erkenntnisse pflichte ich Thomas B. Sheridan (2000) bei, wenn er sagt, dass die verfügbare normative Entscheidungstheorie eine solide Basis für eine Theorie darstelle, die etwas darüber besagt, wie Menschen tatsächlich entscheiden.

12.7 Sicherheitsspezifikation und Bedienfehler im Licht des Risikos

Ein *Bedienfehler* ist als bedienerseitiger Verstoß gegen die Spezifikation definiert. Unter dem Aspekt des Risikos von Bedienhandlungen lässt sich der Begriff weiter präzisieren: Unter einem *Bedienfehler im weiteren Sinn* ist eine durch den Bediener verursachte und prinzipiell vermeidbare Erhöhung des Risikos zu verstehen.

In der Praxis wird wohl nicht jede geringfügige Erhöhung des Risikos gleich als Bedienfehler gebrandmarkt. In der Sicherheitstechnik gelten Anlagen und Zustände als sicher, wenn das Risiko unterhalb eines gewissen Grenzrisikos bleibt. Wir haben es also mit drei Definitionen des Bedienfehlers zu tun. Ein Bedienfehler ist

1. ein bedienerseitiger Verstoß gegen die Spezifikation (Bedienfehler im engeren Sinn),
2. eine vermeidbare Erhöhung des Risikos durch den Bediener (Bedienfehler im weiteren Sinn) oder
3. eine bedienerverursachte Überschreitung des Grenzrisikos (sicherheitsbezogener Bedienfehler).

Das Normen- und Richtlinienwerk geht davon aus, dass für sicherheitstechnisch relevante Einrichtungen ein Grenzrisiko implizit oder explizit gegeben ist. Die Sicherheitsspezifikation einer Einrichtung soll sicherstellen, dass bei zulässigen Eingaben das Risiko unterhalb des

Grenzrisikos bleibt. Ein *sicherheitsbezogener Bedienfehler* ist demnach nichts anderes als ein bedienerseitiger Verstoß gegen die Sicherheitsspezifikation. Im sicherheitstechnischen Sinn bedeuten die 1. und die 3. Definition demnach das Gleiche.

12.8 Entscheiden nach Faustregeln

Mit dem Entscheidungsmodell auf der Grundlage des subjektiven Risikos steht uns so etwas wie eine Richtschnur zur Verfügung: Es sagt, wie wir vernünftigerweise entscheiden sollten. Abweichungen davon sind *Fehlentscheidungen*. Insofern ist dieses Entscheidungsmodell normativ.

Unsere Entscheidungsprozesse gehorchen oft nicht dem normativen Modell. Es gibt systematische Abweichungen vom rationalen Verhalten. Gründe für Fehlentscheidungen können sein:

- das Übersehen relevanter Entscheidungsalternativen (unvollständiger Entscheidungsbaum),
- ein falsches Bild der Konsequenzen (fehlerhafter Ereignisbaum),
- Fehleinschätzung von Nutzen und Kosten der Konsequenzen,
- Fehleinschätzung der Ereigniswahrscheinlichkeiten,
- übersteigerte Risikoakzeptanz oder Risikoaversion.

Die Kognitionspsychologie zeigt typische Situationen auf, in denen es zu Irrtümern kommt. Es empfiehlt sich, diese Fälle zu studieren. Dabei lernt man die Warnzeichen von Denkfallen kennen. Ist eine Denkfalle erst einmal als solche erkannt, lässt sich der Reinfall vermeiden. Die wichtigsten Denkfallen von Entscheidungssituationen bringt das populärwissenschaftliche Buch von Hammond, Keeney und Raiffa (1999). Ein Klassiker auf diesem Gebiet ist die von Kahneman, Slovic, und Tversky herausgegebene Sammlung (1982). Eine Auswahl ist in meinem Aufsatz über Bedienfehler enthalten (1998).

Wie leicht wir Nutzen oder Kosten einer Entscheidung falsch einschätzen, zeigt das *Paradoxon von Braess* (Spektrum der Wissenschaft, 1992, Heft 11, S. 23-26). Das Paradoxon entsteht in Entscheidungssituationen, wie sie beispielsweise einem Stadtplaner begegnen: Soll die Entlastungsstraße gebaut werden oder nicht?

Ich erläutere das Paradoxon an einem Zweipersonenspiel. Vorgelegt wird vom Spielleiter die Zeichnung eines Rechtecks. Jeder Spieler hat die Aufgabe, entlang der Kanten dieses Rechtecks einen Weg zu suchen, der ihn von der linken oberen Ecke zur rechten unteren Ecke führt, und zwar zu möglichst geringen Kosten. Die Wegekosten betragen 2 Einheiten für jede der horizontalen und 5 Einheiten für jede der vertikalen Kanten. Wird eine der Kanten von beiden Spielern gewählt, haben sie - wegen gegenseitiger Behinderung - den doppelten Preis zu zahlen, Bild 12.3.

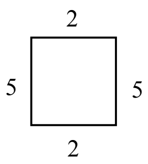


Bild 12.3 Wegeplan (Version 1)

Einer der Spieler wählt seinen Weg „oben herum“ und zahlt die Wegekosten 7 (=2+5). Der andere geht „unten herum“ und zahlt ebenfalls 7 (=5+2) Einheiten.

Der Spielleiter eröffnet nun den Spielern die Möglichkeit, ihre Kosten zu senken, indem er eine weitere Verbindung von rechts oben nach links unten einführt. Diese Verbindung ist kostenlos, Bild 12.4.

Diese Möglichkeit nutzt einer der Spieler auch tatsächlich aus. Er zahlt nun nur noch 6 (=2+0+2·2) Einheiten. Da eine der Verbindungen von beiden benutzt wird, muss der andere Spieler nun mehr zahlen, nämlich 9 (=5+2·2) Einheiten. Das lässt ihm keine Ruhe, und er macht es wie sein Gegenspieler. Beide wählen schließlich den z-förmigen Weg und zahlen jeweils 8 (=2·2+2·0+2·2) Einheiten. Jeder der Spieler kann nur noch zu seinem eigenen Nachteil von der schließlich getroffenen Entscheidung abweichen. Dumm ist nur, dass beide jetzt schlechter fahren als zu Beginn, als es die Entlastungsverbindung noch nicht gab. Aus dieser misslichen Situation kommen die Spieler nur heraus, wenn sie beschließen, gemeinsam den Entlastungspfad (!) zu ignorieren und zu ihren ursprünglichen Wegen zurückkehren.

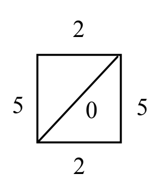


Bild 12.4 Wegeplan (Version 2)

Wie misslich die Situation ist, wird vollkommen deutlich, wenn wir uns die Spielmatrix dieses Spiels ansehen (Tabelle 12.3). Es handelt sich nämlich um eine Spielart des *Gefangenen-Dilemmas* (Mérö, 1998). Es gibt im Prinzip zwei Verhaltensweisen: Kooperation K und deren Negation N. Der kooperierende Spieler meidet die kostenlose Verbindung. Der andere nutzt sie.

Tabelle 12.3 Spielmatrix

		Aktion des Gegenüber	
		K	N
Kosten der Aktion	K	7	9
	N	6	8

Die Spielmatrix zeigt die Kosten der Aktion jeweils für den Zeilenspieler. Wer sich nicht auf den anderen verlassen kann, tut gut daran, die Negationsentscheidung zu treffen. Egal was der andere tut, er ist in jedem Fall besser dran als mit der Kooperationsentscheidung.

Zurück zum Planungsproblem des Stadtplaners: Da Verkehrsteilnehmer nicht auf Kooperation eingestellt sind, wird die Entlastungsstraße die Verkehrssituation wohl verschlimmern. Hätten Sie das gedacht?

Manche Fehlentscheidung geht wohl darauf zurück, dass wir Konsequenzen nicht gut genug durchdenken. Anstelle einer sorgfältigen Analyse entscheiden wir nach *Heuristiken (Faustregeln*). Das geht schneller und entlastet den Denkapparat. Eine der Faustregeln unseres Verkehrsplaners könnte lauten „Entlastungsstraßen entlasten". Das Paradoxon zeigt, dass er mit der Faustregel daneben liegen kann.

Manchmal geht es nicht ohne solche Faustregeln, weil die Zeit für eine gründliche Analyse gar nicht da ist, und weil in vielen Fällen eine nur plausible Entscheidung immer noch besser ist als gar keine. Oder wie ein Industriemanager einmal sagte: „Der Fortschritt lebt vom Mut der Ahnungslosen". Noch allgemeiner gesprochen: Faustregeln haben ihren Nutzen, sonst hätten sie sich im Laufe der Evolution nicht durchsetzen können. Die Mechanismen sind schon in Ordnung, aber manchmal eben fehl am Platze.

Es geht hier nicht darum, Faustregeln zu ächten, sondern darum, dass *der kluge Entscheider ein Gefühl für die Grenzen seiner Faustregeln entwickelt. Und er tut das über das Studium von Denkfallen*.

12.9 Gefährliche Strukturen

Die obige Fallsammlung enthält - neben weniger schwerwiegenden - auch einige Fehler, die zu Katastrophen geführt haben. Man wird diesen Fällen nicht gerecht, solange man nur personenbezogene Fehlermechanismen und allein die Fehlentscheidungen von Einzelpersonen in Rechnung stellt.

Normale Katastrophen

Eine monokausale Betrachtung von Unfallhergängen ist grundsätzlich verkehrt. An einem Unfall ist normalerweise ein ganzes Bündel von Ursachen beteiligt. In Systemen mit vielen und untereinander eng gekoppelten Komponenten und bei hohem Gefährdungspotential haben die Wirkungslinien der vielen kleinen oder gar alltäglichen Ursachen Gelegenheit, tatsächlich irgendwann einmal unglücklich zusammenzutreffen und zu einer Katastrophe zu führen. Charles Perrow (1999) spricht in diesem Zusammenhang von *normalen Katastrophen*.

Katastrophen gehen meist Entscheidungen voraus, die im Nachhinein als Fehlentscheidungen erkennbar sind. Und diese Entscheidungen werden nicht von Einzelpersonen, sondern von Kollektiven und Organisationen getroffen. Um solche organisatorisch und sozial bedingten Fehlentscheidungen geht es in den oben angesprochenen Fällen des Challenger-Unglücks und der Katastrophe von Eschede.

„The Challenger Launch Decision“ von Diane Vaughan (1996) behandelt das Challenger Unglück ausführlich: Die Verbindungen zwischen den Raketensegmenten der Feststoffraketen waren von der Organisation NASA lange vor dem Unglücksflug als kritisch erkannt worden. Speziell den etwa 6 Millimeter starken Dichtungsringen galt die Aufmerksamkeit. Ein solcher Dichtungsring umfasst die gesamte Feststoffrakete und hat eine Länge von über elf Metern; die Qualitätsanforderungen sind hoch. Entwickelt sich nach gezündeter Rakete Druck, dann wird der Spalt an der Verbindungsstelle zwischen den Raketenteilen breiter und die Dichtung muss sich ausdehnen. Die Dichtungsringe sind für die zur damaligen Startzeit herrschende niedrige Temperatur nicht ausgelegt: je kälter es wird, desto langsamer dehnen sie sich aus. Zusammen mit weiteren Ursachen, beispielsweise durch Drucktests verursachte Verschleißerscheinungen an den Ringen, kann es zum Versagen eines Dichtungsrings kommen.

Dieses Risiko war bekannt und wurde auch in den entscheidenden Sitzungen vor dem Start zur Sprache gebracht. Ein wesentliches Argument für den Start war der Hinweis auf die Redundanz: jede Verbindung wird nämlich - anders als bei der Vorläuferrakete Titan - durch zwei Ringe abgedichtet. Aber bereits vorher war die Wirksamkeit dieser Redundanz in Zweifel gezogen worden. Die Notwendigkeit, zwei Ringe unterzubringen, machte eine andere Geometrie der Verbindung nötig. Und eine Analyse hatte gezeigt, dass der zweite Ring bei dieser Spaltgeometrie unter Umständen gar nicht wirksam wird. Daraufhin hatte man die Verbindung nicht länger als redundant klassifiziert. Nach wie vor aber galt den Entwicklungsteams und Managern die Redundanz im Regelfall als wirksam. Das mit der Verbindungsstelle verbundene Risiko wurde als akzeptabel eingestuft.

Zum Startzeitpunkt herrschte eine Außentemperatur, für die das System nicht ausgelegt und auch nicht erprobt war. Der Start unter diesen Bedingungen kann als anwenderseitiger Verstoß gegen die „Spezifikation“ aufgefasst werden, als Bedienfehler also - auch wenn das in diesem Zusammenhang sonderbar klingt.

Die soziale Konstruktion von Risiko in Organisationen

Die Bedenken einiger Ingenieure gegen die Challenger-Mission wurden im entscheidenden Moment durch den Hinweis auf die Redundanz zerstreut. Dieser *Glaube an die Redundanz* und die *Sicherheitserfahrung* mit den vorhergehenden Flügen (einschließlich denen der Titan-Rakete) führten zur Unterschätzung des Risikos. Im Nachhinein wurden die Häufigkeiten von Lecks in Abhängigkeit von der Temperatur für sämtliche Flüge ermittelt. Dabei stellte sich heraus, dass bei den zur Startzeit herrschenden Temperaturen ein Versagen der Dichtungen hoch wahrscheinlich war. Und bei einer Versagenswahrscheinlichkeit nahe eins ist Redundanz von vornherein unwirksam, denn aus $p \approx 1$ folgt $p^2 \approx 1$.

Der Glaube an die Redundanz und die Sicherheitserfahrung gehen auf die Tendenz zur *Überbewertung bestätigender Informationen* zurück. Dieser Mechanismus verführt Experten dazu, Informationen zu übersehen, die einer Überzeugung widersprechen (Vaughan, 1996, S. 237; Feynman, 1988, S. 216 ff.). Hier ist es die Überzeugung, dass das System sicher ist.

Serienteile - wie diese Dichtungsringe - werden bei innovativen Systemen gelegentlich unter Bedingungen eingesetzt, die von deren Spezifikation nicht abgedeckt sind. An Stelle des Industriestandards treten dann „hausinterne" Regeln. Nach diesen Regeln werden Risiken *definiert*. Die Verfahren, nach denen solche Regeln entstehen, sind Bestandteil der Unternehmenskultur. Es sind demnach die Banalitäten der Bürokratie, die es erlauben, dass Abweichungen von den Sicherheitsvorschriften zur Gewohnheit werden.

Wir haben hier alle Merkmale beisammen, die für eine soziale - also gemeinschaftlich unternommene - Konstruktion des Risikos sprechen. Für eine *Organisation*, deren Ziel die Entwicklung und die Anwendung einer Technik ist, lassen sich einige der dabei wirksamen Mechanismen identifizieren (Hendrick, 1997; Luhmann, 1991, S. 204 ff.; Douglas/Wildavsky, 1982, S. 100; Douglas, 1986; Perrow, 1999, S. 378 ff.; Leveson, 1995, S. 53 ff.):

1. Horizontale Differenzierung, die Gliederung in Abteilungen, schafft Gruppen von Spezialisten mit der Tendenz zur Ausbildung *spezieller Risikokulturen*.
2. Vertikale Differenzierung schafft hierarchische Strukturen mit der unvermeidbaren Folge der *Filterung* von Informationen aufwärts zum Entscheider und abwärts zum Ausführenden. („Geben sie mir die Kennzahl zur Sicherheit des Systems - mit ihrer Unterschrift. Die Wenns und Abers interessieren nicht.")
3. Die *Formalisierung* von Abläufen und Entscheidungsprozessen ist eine Form, wie Organisationen erworbenes Wissen festhalten, kurz: lernen. Eine solche Formalisierung ist die von der NASA verwendete Critical Items List (CIL). Sie klassifiziert Komponenten nach dem Schadensausmaß bei einem Versagen. Die Einordnung von sicherheitstechnischen Systemen in Anforderungsklassen (Abschnitt 6.1) ist ein ähnliches Klassifizierungssystem. Nicht Messbares wird im Konsensverfahren innerhalb einer Gruppe festgelegt; Risiko wird nicht ermittelt, es wird konstruiert.
4. *Bürokratisches Verhalten* ist als risikoavers und konservativ bekannt. Es gilt, Änderungen zu meiden und Überraschungen nicht zuzulassen („Change is Bad"). Andererseits wird dadurch die Risikominderung über ein verändertes Design erschwert oder gar verhindert.
5. Es gibt *blinde Flecke*: Risiken werden übernommen, nur weil man sie nicht sehen kann (Douglas/Wildavsky). Formalisierung, Filterung und Ausblendung von Informationen erlauben eine „Beruhigung im Kleinformat". Es gibt Regeln und Formulare, die „laufendes Entscheiden provozieren, aber mit dem gleichen Instrument auch verhindern, dass man über den Rand hinausblickt und Ungewöhnliches bemerkt" (Luhmann).
6. Vertikale und horizontale Differenzierung ziehen die *Sequenzierung* von Entscheidungen nach sich. Folgen von Teilentscheidungen führen allmählich zu irreversiblen Festlegungen.
7. Es sind nicht allein die „Normalisierung der Regelverstöße, die Banalitäten der bürokratischen Abläufe und Hierarchien oder das Ergebnis einer Ingenieur-Kultur", sondern auch die *Machtausübung* innerhalb der Organisation, die zu einer Korrumpierung der Sicherheitskultur führen (Perrow).

13 Bedienfehler und Bedienbarkeit

13.1 „Menschliches Versagen“ und Zuverlässigkeit

Anfang 1995 brachten die Tageszeitungen unter dem Titel *Unsicherheitsfaktor Mensch* eine Nachricht der Internationalen Zivilluftfahrt-Organisation ICAO: „Als Unsicherheitsfaktor Nummer eins erwies sich auch 1994 wieder der Mensch: Nicht weniger als 31 der 47 Unfälle sind auf menschliches Versagen zurückzuführen und immerhin 16 auf das Wetter... Die europäische Airbus Industrie in Toulouse änderte die automatische Steuerung an den A300-600- und A310-Typen.“

Wir wollen einmal beiseite lassen, dass diese Meldung den Beitrag der Technik zur Unzuverlässigkeit des Gesamtsystems beschönigt, denn: weshalb sollte man die Technik ändern, wenn die Unfälle allein auf Mensch und Umwelt zurückgehen? Bereits im ersten Kapitel habe ich dargelegt, warum der mit Schuldzuweisungen belastete Begriff des „menschlichen Versagens“ hier in Anführungszeichen steht. Gemeint sind eigentlich Bedienfehler. Diese tragen zweifellos zu den Risiken der Technik bei. Sie sind in die Systembewertung einzubeziehen.

Eine naheliegende Idee ist, die für Zuverlässigkeitsbetrachtungen gültige Systemgrenze - die Spezifikation (Bild 1.1) - zu verschieben und den Menschen zu einer „Komponente“ des Analyseobjekts zu machen. Diese auf den Menschen ausgedehnte Zuverlässigkeitsanalyse trägt den Namen Human Reliability Analysis (HRA).

Zu den bekanntesten HRA-Methoden gehört THERP (Technique for Human Error Rate Prediction). Sie wurde in den sechziger Jahren von Alan D. Swain und anderen entwickelt (Salvendi, 1997, S. 167 ff.).

Tabelle 13.1 Wahrscheinlichkeiten für Bedienfehler

Geschätzte HEP	*Fehlertyp*
0.3 %	Allgemeiner Handlungsfehler (Error of Commission). Beispiel: Beschriftung falsch gelesen und deshalb den falschen Schalter betätigt.
1 %	Allgemeiner Unterlassungsfehler (Error of Omission). Beispiel: Vergessen, einen Schalter nach einer Testprozedur wieder auf Normalbetrieb zu stellen.
3 %	Einfacher Arithmetikfehler mit Probe, aber ohne separate Wiederholung des Rechengangs auf einem anderen Papier.
10 %	Inspektionsfehler. Ein ursprünglich begangener Fehler des Operateurs entgeht der Wachsamkeit des Kontrolleurs.
20 ... 30 %	Allgemeiner Fehler unter hohem Stress und schnelle Abfolge gefährlicher Schritte.

Die zentrale Maßzahl der HRA ist die *Human Error Probability (HEP*). Diese Fehlerwahrscheinlichkeit des Menschen hängt von der Aufgabe und manch anderem ab und ist definiert

als die Anzahl der tatsächlich begangenen Fehler bezogen auf die Gesamtzahl von Gelegenheiten, diesen Fehler überhaupt zu machen:

$$\text{HEP} = \frac{\text{Anzahl der begangenen "menschlichen Fehler"}}{\text{Gesamtzahl der Gelegenheiten für diesen Fehler}}$$

Die HEP ist also eine auf das „Teilsystem Mensch" bezogene aufgabenabhängige Versagenswahrscheinlichkeit. In Tabelle 13.1 sind einige der im Buch von Salvendi (1997) aufgeführten Schätzwerte der HEP speziell für Bedienfehler wiedergegeben.

Dass sich aus solchen Zuverlässigkeitsdaten überhaupt aussagestarke Zuverlässigkeitsprognosen gewinnen lassen, wird von den Verfechtern der Methode damit begründet, dass das Verhalten des Menschen auch bei variierenden Situationen weitgehend konstant ist („behavior similarity despite equipment dissimilarity").

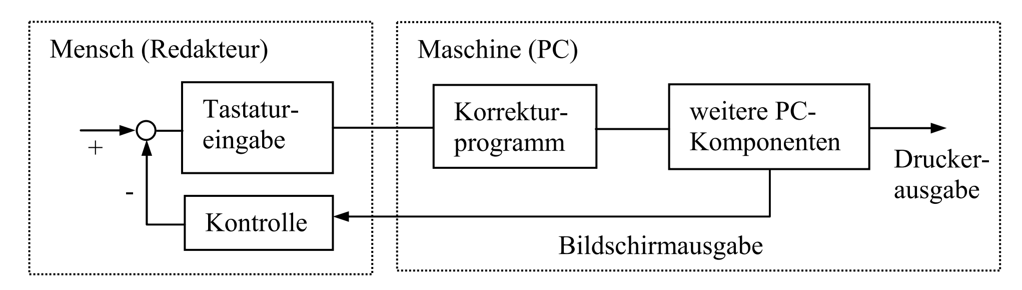


Bild 13.1 Beispiel für ein teilautomatisiertes Mensch-Maschine-System

Als Beispiel für die Analysemethode der HRA nehmen wir ein *Mensch-Maschine-System,* das aus einem Redakteur und seinem Personal Computer mit Textverarbeitungsprogramm und Rechtschreibprüfung besteht. Die Tastatureingabe wird der Rechtschreibprüfung unterzogen und automatisch korrigiert. Das Ergebnis erscheint auf dem Bildschirm und wird im elektronischen Dokument abgespeichert. Der Redakteur kann die Bildschirmdarstellung kontrollieren und den Text gegebenenfalls noch einmal korrigieren. Schließlich wird das fertige elektronische Dokument dem Drucker übergeben. Zur Darstellung des Rückkopplungsprozesses eignet sich die Metapher des Regelkreises, Bild 13.1.

Tabelle 13.2 Die fehlerauslösenden Ereignisse des Mensch-Maschine-Systems mit den (bedingten) Fehlerwahrscheinlichkeiten

Systemkomponente	***Ereignis***	***Wahrscheinlichkeit***	
		Symbol	***Wert***
Eingabe (Tastatur)	Eingabefehler	p_E	0.05
Automatik (Korrekturprogramm)	Korrektur eines fehlerhaften Wortes misslingt	p_{A1}	0.05
	Korrektes Wort wird verfälscht	p_{A2}	0.01
Kontrolle (durch den Redakteur)	Fehler auf Bildschirm wird übersehen	p_K	0.1
Maschine (Personal Computer)	Transfer-, Codierungs- oder Speicherfehler	p_M	0.001

Als auslösendes Ereignis betrachten wir die Eingabe eines Wortes. Die möglichen Fehlerereignisse sind in Tabelle 13.2 zusammen mit ihren - für dieses Beispiel angenommenen - absoluten bzw. bedingten Wahrscheinlichkeiten erfasst.

Wir setzen voraus, dass die Eingabe in den PC genau dann misslingt, wenn das Korrekturprogramm ein von der Tastatur kommendes fehlerhaftes Wort nicht oder nicht richtig korrigiert oder ein richtig geschriebenes Wort verfälscht und wenn der Redakteur den Fehler auf dem Bildschirm übersieht. Maschinenfehler mögen stets zu einer falschen Ausgabe führen.

Der Ereignisbaum in Bild 13.2 erfasst alle möglichen Ereignisabläufe und deren Wahrscheinlichkeiten bis hin zur Abspeicherung des Wortes im Dokument, also ohne die Maschinenfehler. Die Wahrscheinlichkeit eines Texteingabefehlers ist demnach gleich

$$p_{\text{Texteingabe}} = p_E \cdot p_{A1} \cdot p_K + (1-p_E) \cdot p_{A2} \cdot p_K = 0.0012.$$

Über das Gesamtsystem gesehen liegt ein Fehler genau dann vor, wenn die Texteingabe oder wenn die Druckerausgabe fehlerhaft ist. Unter Einbeziehung der Maschine ergibt sich demnach

$$p_{\text{System}} = p_{\text{Texteingabe}} + p_M - p_{\text{Texteingabe}} \cdot p_M \approx 0.0022.$$

Kritik der Methode: Die Zuverlässigkeitsmodellierung der „Komponente Mensch" geschieht in der HRA nach einem rein behavioristischen Ansatz. Sie beschränkt sich konsequent auf beobachtbare Sachverhalte und verzichtet auf eine *Ursachenanalyse* unter Einbeziehung der geistigen Tätigkeiten. Die Zuweisung sinnvoller Fehlerwahrscheinlichkeiten ist auf sehr situationsspezifische Handlungsschnitzer beschränkt (Reason, 1990).

Eingabe	Automatik	Kontrolle	Konsequenz Dokumentfehler	Pfadwahrscheinlichkeit
$(1, p_E)$	$(1, p_{A1})$	$(1, p_K)$	1	$p_E p_{A1} p_K$
		$(0, 1-p_K)$	0	$p_E p_{A1}(1-p_K)$
	$(0, 1-p_{A1})$		0	$p_E(1-p_{A1})$
$(0, 1-p_E)$	$(1, p_{A2})$	$(1, p_K)$	1	$(1-p_E)p_{A2}p_K$
		$(0, 1-p_K)$	0	$(1-p_E)p_{A2}(1-p_K)$
	$(0, 1-p_{A2})$		0	$(1-p_E)(1-p_{A2})$

Bild 13.2 Ereignisbaum der Texteingabe (1: Fehler liegt vor, 0: Fehler liegt nicht vor)

13.2 Vorstellungen, Wahrnehmung, Denken

Zwei grundverschiedene Zugänge bieten sich dem an, der das Verhalten des Menschen als Entwickler und Bedienungsperson in die Risikoanalysen einbeziehen will: Entweder

1. er beschränkt sich auf die direkt beobachtbaren Handlungsmerkmale und gibt sich mit dem Studium der Fehler aufgrund von *Schnitzern* zufrieden, also Fehlern in ansonsten routinemäßig ablaufenden Handlungen. Das ist der behavioristische Ansatz des vorigen Ab-

schnitts. Sein Vorteil ist, dass sich bei einem solcherart stark eingeschränkten Anwendungsbereich das Verhalten des Menschen sogar auf Maßzahlen wie die menschliche Fehlerwahrscheinlichkeit (HEP) abbilden lässt. Oder

2. er versucht, die anspruchsvollen geistigen Tätigkeiten und die dabei vorkommenden *Irrtümer* (also Fehler auf höheren kognitiven Ebenen) in seine Studien einzubeziehen. Damit betritt er angesichts der Komplexität des Untersuchungsgegenstands schwankenden Boden. Die Aussagen haben nicht mehr die Stringenz, die der Ingenieur von Theorien erwartet. Dennoch hat diese mentalistische Betrachtungsweise großen Nutzen: Wir können aus der Analyse der kognitiven Fehlerursachen Regeln gewinnen, die uns bei der Konstruktion besser bedienbarer und zuverlässigerer Einrichtungen leiten. Der Erfolg bei Anwendung der Regeln spricht für die Methode, er wiegt den Mangel an Theoriehaftigkeit auf.

Der zweite Ansatz geht mit einem radikalen Wechsel des Blickwinkels einher. Ich beginne mit dem Paradoxon der *Brains in Vats* (Poundstone, 1988): Sie denken vielleicht, dass Sie jetzt dieses Buch lesen. Aber genausogut könnten Sie ein körperloses Gehirn in irgendeinem Labor sein. Es befindet sich in einem Bottich mit Nährflüssigkeit und ist über viele Elektroden an einen Computer angeschlossen. Ein Wissenschaftler hat ein Experiment programmiert, das genau die Erfahrung, dieses Buch zu lesen, in Ihrem Gehirn erzeugt. Sie leben in einer vollkommen *simulierten Welt*. Und diese Welt ist in ihrem Kopf. Der Computer sorgt nur für die richtigen Stimuli wie beispielsweise die taktilen Signale, die Ihnen das Gefühl geben, eine Seite umzublättern, oder die Sehnervensignale, von denen Sie glauben, sie kämen von ihrer Netzhaut. Vielleicht sind Sie nur Teil der „Matrix". So heißt die Kombination aus Computern und Gehirnen im gleichnamigen Film (1999).

Sie sagen: das kann nicht sein! Aber haben Sie Beweise? Nein, und genau das ist das Paradoxe an der Geschichte. Wenden wir uns nun dem wissenschaftlich nachweisbaren Gehalt dieser phantastischen Geschichte zu.

Klinische Befunde (Sacks, 1987) und die Untersuchungen der Wahrnehmungspsychologen (Goldstein, 1997, S. 181 ff.) belegen: Die von uns wahrgenommene Welt ist eine Konstruktionsleistung unseres Gehirnes. Der Informationsfluss, den unser bewusstes Denken und das Kurzzeitgedächtnis bewältigen können (es sind weniger als 100 Bits je Sekunde) reicht bei weitem nicht aus, um uns die gesamte uns umgebende Welt immer wieder neu bewusst zu machen. Das Allermeiste, was wir zur Orientierung in dieser Welt benötigen, müssen wir bereits *wissen*. Ein Teil des Wissens ist angeboren, aber einen Großteil erwerben wir im ersten Lebensjahr.

Beispiel 1: Erblindete Kinder konnten nach Erreichung des Schulalters operiert werden. Der optische Teil des Auges wurde weitgehend in Stand gesetzt. „Nur wenige erlernten mühsam, einfache Muster zu erkennen ... Viele brachen die Behandlung ab und zogen es vor, als Blinde weiterzuleben. Das Gehirn dieser Patienten blieb unfähig, die von den Augen wieder einwandfrei übermittelten Signale zu verarbeiten ... Was sich im Erwachsenenalter wahrnehmen lässt, hängt also ganz entscheidend von der Art frühkindlicher Erfahrung ab" (Singer, 1985).

Beispiel 2: Erzeugung einer *Scheinbewegung* (Goldstein, 1997, S. 273). Für das kleine Experiment bringen Sie auf einander gegenüberliegenden Seiten eines Streichholzes zwei Punkte so an, dass sie etwas gegeneinander verschoben sind (Bild 13.3). Wenn Sie das Streichholz zwischen den Fingern mäßig schnell hin und her drehen, nehmen Sie die Bewegung *eines* Punktes wahr. Diese Bewegung ist vom Gehirn *konstruiert*, es ist eine Scheinbewegung. Der Eindruck einer Bewegung entsteht nicht, wenn die Drehung zu langsam oder zu schnell geschieht.

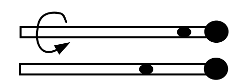


Bild 13.3 Experiment zur Bewegungswahrnehmung

Wer meint, alles was er sieht, nehme er auch wahr, der möge einmal zählen, wie oft der Buchstabe F in dem folgenden Text vorkommt.

FINISHED FILES ARE THE RE- SULT OF YEARS OF SCIENTIF- IC STUDY COMBINED WITH THE EXPERIENCE OF YEARS

Es sind drei, meinen Sie, oder vier, oder fünf? Nein, der Buchstabe F kommt sechsmal vor. Wir übersehen bei der ersten Begegnung mit diesem Text vorzugsweise die Fs im Wörtchen OF. Für den hier zu beobachtenden Effekt wird die *Einheitenhypothese* von A. F. Healy verantwortlich gemacht: OF ist ein Wort, das in der Sprache mit extrem hoher Häufigkeit auftritt und deshalb besonders leicht als eine Einheit gelesen und nicht erst aus Buchstaben zusammengesetzt wird. James Reason (1990, 1994) spricht in diesem Zusammenhang auch von einer *Tarnung durch Vertrautheit*.

Die *Welt in unserem Kopf* hat viel mit einem *Simulationsmodell* gemeinsam. Und dieses Modell lässt sich abkoppeln vom Aktuellen. Wir können mit den Komponenten des Modells experimentieren, uns selbst im vorgestellten Raum bewegen und Dinge neu kombinieren.

Albert Einstein beschreibt sein *Denken in Vorstellungen* so: „Die geistigen Einheiten, die als Elemente meines Denkens dienen, sind bestimmte Zeichen und mehr oder weniger klare Vorstellungsbilder, die 'willkürlich' reproduziert und miteinander kombiniert werden können ... Dieses kombinatorische Spiel scheint die Quintessenz des produktiven Denkens zu sein“ (Krech/Crutchfield, 1992, Band 4, S. 109).

In Form von Thesen stelle ich die grundlegenden Bedingungen zusammen, denen unsere Wahrnehmung und unser Denken unterworfen sind. Die Darstellung der Gestaltwahrnehmung und die Mehrebenenmodelle der folgenden Abschnitte werden vor diesem Hintergrund verständlich. Hinweise auf Quellen und vertiefende Darstellungen sind in meinem Buch „Denkfallen und Programmierfehler“ zu finden.

1. Es gibt einen *Engpass der Wahrnehmung*: Aus den Umweltreizen werden unter Steuerung der Aufmerksamkeit nur kleine Teile selektiert, verarbeitet und bewusst wahrgenommen.
2. Wir nehmen in erster Linie das wahr, wofür wir uns interessieren und wovon wir bereits eine gewisse Vorstellung haben.
3. Wahrnehmung und Erkenntnis sind *erwartungsgetrieben*. Karl R. Popper nennt den Engpass der Wahrnehmung zusammen mit der erwartungsgetriebenen Erkenntnis das *Scheinwerfermodell der Erkenntnis*.
4. Unsere Vorstellungen von der Welt lassen sich mit einem *Simulationsmodell* vergleichen. Es ist weniger eng mit der uns umgebenden Welt gekoppelt als wir meinen.
5. *Sinnsuche des Wahrnehmungsapparats*: Nur wenige Hinweise und Signale genügen, um unser Wissen zu aktivieren und damit komplexe Vorstellungsbilder zu konstruieren, die sich in ein widerspruchsfreies *mentales Modell* integrieren lassen.
6. Die Güte des mentalen Modells (gemessen am Erfolg unserer Handlungen) hängt davon ab, inwieweit die Wahrnehmungsfilter die richtigen Reize selektieren und verarbeiten und inwieweit der Scheinwerfer der Erkenntnis richtig ausgerichtet ist.
7. *Irrtümer* drohen, wenn mentales Modell und Wirklichkeit nicht zusammenpassen.

13.3 Die Gestaltgesetze der Wahrnehmung

Wer fehlerfrei bedienbare Geräte bauen will, muss zuallererst die grundlegenden Wahrnehmungsmechanismen kennen und diese für die Unterstützung des Bedieners nutzen. Wir lassen die Wahrnehmungspsychologen zu Wort kommen (Goldstein, 1997).

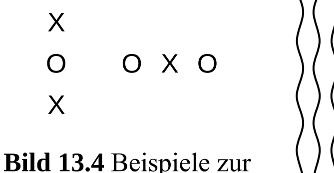
Alles Leben geht augenscheinlich von der Hypothese eines objektiv existierenden Kosmos aus, der von Recht und Ordnung zusammengehalten wird. Diese *Strukturerwartung* hat sich im Laufe der Evolution als Erfolgsrezept erwiesen. In der optischen Wahrnehmung kommt sie in den *Gestaltgesetzen* zum Ausdruck. Zu den wichtigen Gestaltgesetzen gehören die folgenden (Bild 13.4):

Prägnanztendenz: Der Wahrnehmungsapparat sucht stets nach Zusammenhängen größtmöglicher Einfachheit und Regularität.

Gesetz der Nähe: Dinge, die nahe beieinander liegen, scheinen zusammenzugehören.

Gesetz der Ähnlichkeit: Ähnliche Dinge erscheinen zu zusammengehörigen Gruppen geordnet.

Figur-Grund-Trennung: Die Reizmuster eines Bildes verdichten sich in unserer Wahrnehmung meist zu Gegenständen, die von ihrem Hintergrund getrennt erscheinen. In den Kippbildern bleibt unentschieden, was Hintergrund und was Gegenstand ist. Dennoch wird getrennt: mal so, mal anders. Der „Lattenzaun" ist ein Beispiel dafür. Die Latten erscheinen oben konkav oder konvex gewölbt, je nachdem ob man den Zaun von links nach rechts oder von rechts nach links durchmustert.



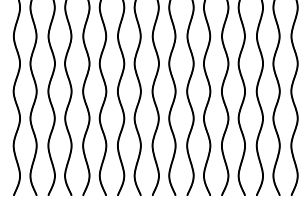


Bild 13.4 Beispiele zur Gestaltwahrnehmung

13.4 Taxonomien der Denk- und Handlungsfehler

Es gibt eine Fülle von Ansätzen, die kognitiven Fehlerursachen systematisch zu erfassen und in eine Fehlertaxonomie zu bringen. Beispiele:

- der Action Cycle von Norman (1988),
- die Heuristiken von Kahneman und Tversky (1974),
- die Liste der Fehlerursachen beim Planen und Entscheiden von Dörner und Schaub (1994),
- die Klassifizierung der Schnitzer von Norman (1981),
- das Selective Scrutiny Model von Evans (1989),
- das Generische Fehlermodellierungssystem (Generic Error-Modelling System, GEMS) von Reason (1994),
- das Skill-Rule-Knowledge-Modell von Rasmussen (Reason, 1994).

Das Generische Fehlermodellierungssystem von Reason ist eines der Mehrebenenmodelle der Denk- und Handlungsfehler. Es geht davon aus, dass der Mensch „ein Gewohnheitstier" ist: Im Sinne des Sparsamkeitsprinzips schenken wir uns das Denken, wenn wir bereits mit Routine und automatisierten Abläufen zurechtkommen (fähigkeitsbasierte Ebene). Und wir verzichten auf logische Schlussfolgerungen, wenn ein Sachverhalt glaubwürdig erscheint. Kurz: Wir ver-

suchen unsere Alltagsaufgaben und -probleme auf möglichst niedriger kognitiver Ebene zu lösen (Kasten „Ablaufplan des Generischen Fehlermodellierungssystems (GEMS)“).

Unsere Schwierigkeit besteht darin, ein Problem zu erkennen, solange wir der Routine verhaftet sind. Woher nämlich wollen wir auf der niedrigeren Ebene erfahren, dass eigentlich höhere Anstrengungen erforderlich sind? Gerade auf der automatisierten Ebene kommt es zu Fehlern in der Aktivierung von Handlungsschemata aufgrund einer Verkennung der Situation, Fehlinterpretation von Hinweisen, Vergessen von Handlungsschritten, Überlagerung des geplanten durch ein gewohntes Schema („Er ging in das Schlafzimmer, um sich für das Abendessen umzuziehen - und fand sich im Bett wieder“). Donald Norman (1981, 1988) hat sich mit genau dieser Sorte von Fehlern, den *Schnitzern,* näher befasst.

Wir verhalten uns oft wie die „Maus, die ihren Weg gelernt hat“: „Entfernte ich aus der Bahn meiner Wasserspitzmäuse ein erhabenes Hindernis, auf das hinaufzuspringen und auf dem weiterzulaufen sie gewohnt waren, so sprangen sie an der betreffenden Stelle in die leere Luft und blieben dann zunächst desorientiert auf dem Boden sitzen... Dann begannen sie schnurrhaartastend zu explorieren, wandten sich rückwärts und erkannten dann sichtlich ein Wegstück wieder, das sie eben, vor der Störung, durchlaufen hatten. Nun faßten sie neuen Mut, wandten sich in die vorherige Richtung, sausten los - und sprangen an der kritischen Stelle noch einmal ins Leere!“ (Lorenz, 1973).

Ablaufplan des Generischen Fehlermodellierungssystems (GEMS)

```
FALLS kein Problem bemerkt wird DANN
   bleibe auf der Ebene der Routine (skill based level)
ANSONSTEN WIEDERHOLE
   FALLS das wahrgenommene Muster bekannt ist DANN
      wende eine der bekannten Regeln an (rule based level)
   ANSONSTEN
      gehe auf die wissensbasierte Ebene (knowledge based level),
      analysiere die Beziehungen zwischen Struktur und Funktion,
      leite eine Diagnose ab und formuliere die Korrekturhandlungen,
      handle,
      und überwache das Ergebnis
BIS das Problem gelöst ist
```

Besonders interessant sind Irrtümer oder Schnitzer, die unter vergleichbaren Umständen immer wieder und bei verschiedenen Personen auftreten, die sich also auf überindividuelle Verhaltensdispositionen zurückführen lassen. Dann lässt sich etwas Verallgemeinerbares daraus lernen. Unter diesem Aspekt betrachten wir die *Ursachen typischer Bedienfehler:*

1. Situationsfaktoren wie
   - ungeeignete Arbeitsplatzgestaltung und schlechte Umgebungsbedingungen
   - Informationsüberlastung
2. physische, emotionale, soziale und organisatorische Faktoren wie
   - Müdigkeit, Krankheit und Stress
   - kommunikationsfeindliches und demotivierendes Betriebsklima
   - schlechte Ausbildung
3. Gefangennahme der Aufmerksamkeit (Scheinwerferprinzip)
4. Wahrnehmungsfehler
5. Automatisierung des Denkens und Handelns sowie die dabei auftretenden Schnitzer

6. Selbstzufriedenheit bei unbewusster Gefahr (Complacency)
7. Fehlentscheidungen aufgrund von
   - übertriebener Risikoakzeptanz
   - kognitiven Täuschungen (Denkfallen)

Die Automatisierung der Denkvorgänge beruht darauf, dass uns frühere Erfahrungen dazu verleiten, beim Lösen eines Problems bestimmte Denk- und Handlungsweisen (Operatoren) gegenüber anderen vorzuziehen (Anderson, 1988, S. 210 ff.). Das blinde Wiederholen von früher erworbenen Reaktionsmustern entlastet den Denkapparat. Es kann aber auch das Lösen von Problemen erschweren. Es besteht die Tendenz, in einen Zustand der Mechanisierung - oder: *Einstellung* - zu verfallen (Luchins, 1942). Zu Fehlern kommt es, wenn der Zustand nicht verlassen wird, obwohl dies angezeigt ist.

Ein Beispiel: Ein Stahlseil hatte sich bei Arbeiten in einem Lager derartig zu einer engen Schleife zusammengezogen, dass es mit Muskelkraft nicht mehr auseinanderzuziehen war. Die rettende Idee, einen kleinen Kran zu Hilfe zu nehmen, kam zunächst niemandem in den Sinn - denn Kräne sind erfahrungsgemäß zum Heben von Lasten da und nicht zum Lösen von Knoten. So etwas nennt K. Duncker *funktionale Gebundenheit:* Die Gebrauchsanleitung eines Gegenstands scheint mit diesem fest verknüpft zu sein. Man kann es auch *objektorientiertes Denken* nennen (Grams, 1992).

Hat sich ein Operator oder eine Regel bei der Lösung bestimmter Aufgaben und Probleme in der Vergangenheit gut bewährt, dann wird man in Situationen, die ähnliche Merkmale aufweisen, darauf zurückgreifen. Zeichnet sich eine Problemlage weitgehend durch die besagten Merkmale, aber auch durch einige abweichende aus, dann kann es zur falschen Anwendung der 'guten Regel' kommen, kurz: Die Regel ist *bewährt aber falsch (strong but wrong)*. Je geübter jemand darin ist, eine bestimmte Aufgabe auszuführen, desto wahrscheinlicher werden seine Fehler die Form 'bewährt aber falsch' annehmen (Reason, 1994, S. 87).

In diesem Kapitel geht es um diejenigen Bedienfehler, an denen das Bewusstsein nicht oder nur am Rande beteiligt ist. Diese Fehler gehen auf die Gefangennahme der Aufmerksamkeit (3), Wahrnehmungsfehler (4), Routine (5) und Selbstzufriedenheit (6) zurück. Sie stehen in unmittelbarem Zusammenhang mit der Gestaltung der Bedienoberfläche: Eine schlechte Bedienoberfläche kann Fehler dieser Gruppen provozieren - eine gute hilft, sie zu vermeiden. Bei vollem Bewusstsein erfolgte Bedienfehler werden unter dem Stichwort Fehlentscheidung abgehandelt. Fehlentscheidungen gehen vor allem auf zu große Risikobereitschaft und auf Fehldiagnosen zurück. Und davon war schon im letzten Kapitel die Rede.

13.4 Fallsammlung von Bedienfehlern

Fast jeder hat sich schon über unbedienbare Diaprojektoren oder Videogeräte geärgert. Die Reklamation beim Verkäufer führt zu nichts: Mehr oder weniger höflich wird einem bedeutet, dass man eben den Herausforderungen der modernen Welt nicht so recht gewachsen ist.

Dabei liegen die Ursachen der Probleme nicht beim Kunden. Schuld sind oft schlechtes Design, eingebaute Denkfallen, Bedienungsanleitungen, die dem Kunden die Schuld am schlechten Nutzwert zuschieben, Handbücher, deren Zweck nur darin besteht, den Hersteller von Produktmängeln zu entlasten, und so weiter. Die folgenden Beispiele sind großteils aus meiner Privatsammlung. Ich gebe sie in dem Erzählton wieder, in dem sie mir berichtet worden sind oder wie ich eigene Erlebnisse notiert habe.

Manchem Fehlerfall ist eine Ursachenanalyse angeschlossen. Darin werden die Fehler den oben klassifizierten Fehlerursachen zugeordnet. Diese Zuordnung ist - wie bereits erwähnt -

nicht zwingend. Andere Deutungen sind nicht ausgeschlossen. Die Ursachenanalysen sollen Anhaltspunkte für Design-Regeln der Bedienbarkeit liefern, nicht mehr und nicht weniger.

Fall 1: Warnungen sind harmlos. Im Praktikum zur Einführung in die Programmierung stellt eine Gruppe ihr Programm vor. Mir fällt ein Unterprogramm auf, das umständlicher als eine früher gefundene Lösung ist. Ich frage: „Warum gerade diese Variante?“ - „Mit der anderen hat das Programm nicht funktioniert.“ - „Wie nicht funktioniert?“ - „Nach dem Start gab es keinerlei Reaktion mehr. Der Computer musste neu gestartet werden.“ Ich bat, das Programm neu zu übersetzen. Der Übersetzer lieferte eine Flut von Warnungen. Aber er schloss die Übersetzung erfolgreich ab. „Solche Warnungen kommen öfter - auch bei anderen Programmen. Sie sind harmlos.“ Das stimmt wohl meistens, dennoch verlange ich: „Wir akzeptieren keine Warnungen.“ Danach wird das Programm Schritt für Schritt verbessert. Die Warnungen verschwinden nach und nach. Übrig bleibt die Warnung „Suspicious pointer conversion“ (verdächtige Zeigerumwandlung). Diese konnte auf einen schwerwiegenden Programmierfehler zurückgeführt werden. Das inkriminierte Unterprogramm hatte damit nichts zu tun.

Ursachenanalyse: Die Fehlbedienung des Übersetzungsprogramms (Compilers) bestand darin, die Warnungen zu ignorieren. Wenn man mit einem Rezept - hier der Missachtung von Warnungen - bisher ganz gut gefahren ist, so führt diese Erfahrung zu einem übertriebenen Vertrauen in dieses Rezept. Es wird zur Routine. Wir haben es mit einem Fehler der Automatisierung des Denkens und Handelns zu tun. In der Psychologie findet sich dafür auch die Bezeichnung des Einstellungseffekts (Anderson, 1995). Auch die *Selbstzufriedenheit bei unbewusster Gefahr* (*Complacency*) kommt als Fehlerquelle in Betracht (Leveson, 1995). Zu den Complacency-Fehlern gehören

- die Missachtung von Warnzeichen,
- ein zu großes Vertrauen in sicherheitserhöhende Maßnahmen wie Redundanz,
- die Annahme eines mit der Zeit abnehmenden Risikos sowie eine
- zu große Risikoakzeptanz - insbesondere im Zusammenhang mit sehr unwahrscheinlichen aber katastrophalen Konsequenzen.

Die Annahme eines abnehmenden Risikos geht auf „Sicherheitserfahrung“ zurück: Bisher ist nichts passiert, also wird auch zukünftig nichts passieren. Dem hält der Physiker Richard P. Feynman entgegen (1988): Trifft beim Russischen Roulett der erste Schuss nicht, lassen sich daraus keine beruhigenden Schlüsse für den zweiten ziehen.

Die Missachtung von Warnzeichen und weitere Complacency-Fehler spielten eine Rolle beim *Therac-25-Desaster,* bei dem es eine Reihe von Todesfällen durch viel zu hohe Strahlendosen bei der medizinischen Behandlung gab. Auch das Unglück von *Bhopal,* bei dem mindestens 2500 Menschen durch ein Gasleck in einer Pestizidfabrik ums Leben kamen, lässt sich vor allem auf Complacency-Fehler zurückführen (Leveson, 1995).

Fall 2: Wasserhähne. In der Toilette einer Gaststätte wollte ich die Hände waschen. Ein Becken war an der Wand, darüber eine elegant geschwungenes Stück Metall. Das muss der Wasserhahn sein, denke ich. Aber wie geht er an? Ich versuche dies und das, drehe ein wenig am Ende des Metallteils herum. Plötzlich schießt Wasser aus dem Ding, aber an einer Stelle, wo ich es nicht vermutet habe.

Ursachenanalyse: Nur der Anbringungsort und die Abwesenheit konkurrierender Gebilde sagt, um was es sich handeln könnte. Stimulierende Signale fehlen. Und wo keine Signale sind, kann nichts wahrgenommen werden. Das mentale Modell des Objekts bleibt vage. Es folgt noch ein Beispiel zum Thema Wahrnehmungsfehler:

Elegantes Schwarz: Das Radio ist in feinem Schwarz gehalten. Die Schrift klein, so dass der elegante Gesamteindruck nicht gestört wird. Sehschwache Leute müssen sich dann mit uneleganten Aufklebern behelfen, wenn sie das Gerät im Alltag überhaupt bedienen wollen. Der Designer ist vermutlich auch noch stolz darauf, dass die Tasten nahezu fugenlos aneinander stoßen, so dass keine Übergänge sicht- und fühlbar werden. Auch das erhöht die Bedienbarkeit des Gerätes nicht.

Fall 3: Fotoapparat. Wenn zu wenig Licht da ist, klappt ein kleiner Blitz automatisch heraus. Die Kamera bietet keine Hinweise, wie man sie halten muss, wenn die Sache funktionieren soll. Ein Bekannter, der die Eigenheiten der Kamera nicht kannte, hielt sie falsch. Nun ist der Klappmechanismus kaputt. Ein Gummi dient seither als Notbehelf.

Ursachenanalyse: Man hat sich angewöhnt, einen Fotoapparat in bestimmter Weise zu halten und zu bedienen. Der Entwickler des Fotoapparats hat keine Rücksicht auf die gewohnten und automatisierten Denk- und Handlungsabläufe genommen.

Fall 4: Web-Angebot. Es gibt Web-Pages, auf denen sind die Stichwörter zu den weiterführenden Links über das Bild eines Puzzles verteilt. Vermutlich soll das demonstrieren, dass die Teile des Angebots ganz wunderbar zusammenpassen. Das Durchmustern solcher nach dem Zufallsprinzip über die Fläche gestreuten Stichwörter ist mühsam und manchmal entgeht einem dabei das Gesuchte.

Ursachenanalyse: Wir sind gewohnt, zeilenweise und von oben nach unten zu lesen. Nach Zeilenende richten wir den Blick gewohnheitsmäßig auf den Anfang der nächsten Zeile, um mit der Suche fortzufahren. Das Hintertreiben dieser Leseroutine ist vielleicht originell, aber nicht bedienerfreundlich.

Fall 5: Typing fast. Dem Computer Risks Forum, Volume 20, Issue 64 (Thursday 4 November 1999) entnehme ich eine Meldung folgenden Inhalts: Bei der Arbeit mit dem Tabellenkalkulationsprogramm Excel verabschiedet sich plötzlich und ohne Vorwarnung das Programm. Nach Neustart ist festzustellen, dass die Arbeit einer ganzen Stunde verloren ist. Als Ursache stellt sich heraus, dass die Panne bei Eingabe des Texts „// cannot be tested in test harness" passiert sein muss. Und das ging so:

- Excel 97 behandelt den Schrägstrich wie das Drücken der Alt-Taste, was den Speicher-Auswahlknopf in der Menü-Zeile aktiviert.
- Der zweite Schrägstrich wird ignoriert.
- Das Drücken der Leertaste öffnet das Drop-Down-Menü.
- Das „c" wählt die „close"-Option des Menüs.
- Darauf öffnet sich die Dialog-Box mit der Auswahl „Do you want to save? Yes / No / Cancel".
- Das „a" wurde ignoriert.
- Das „n" aktivierte die „No"-Option der Dialog-Box.
- Das war's dann. Excel verabschiedet sich ohne Sicherung der Datei.

Ursachenanalyse: Hier kommen wenigstens drei Dinge zusammen.

1. Der Geübte geht zunehmend zu automatisierten Abläufen über und reduziert dementsprechend die Überwachung.
2. Es gibt eine starke Verkopplung der verschiedenen Betriebsmodi (hier: Eingabemodus und Befehlsmodus) durch eine Vielzahl von Möglichkeiten, von einem Modus in den anderen zu wechseln. Die Umschaltung geht mit Tastenkombinationen in mehreren Varianten, über eine tastaturgesteuerte Menüauswahl und mittels mausgesteuerter Menüauswahl. Diese an sich als benutzerfreundlich gedachte Eigenschaft heutiger Bürosoftware erhöht die ver-

deckte Komplexität und führt im Alltag dazu, dass man zuweilen versehentlich den Modus wechselt.

3. Die Rückmeldung über den Moduswechsel und den erreichten Programmzustand ist schwach. Es kommt zu Wahrnehmungsfehlern. Man erfährt nicht, wo man gelandet ist. Die Kontrolle geht verloren, das mentale Modell kann nicht nachgeführt werden.

Das Aufeinandertreffen von Routine und ungenügender Signalisierung sind im Alltag häufig Quellen von Ärgernissen. Ein paar Beispiele:

Multifunktionsknopf - Variation über ein leidiges Thema (siehe Einleitung). Mein Radio funktioniert reibungslos, bis eines Tages, nach dem Einschalten, kein Ton kommt - oder doch nur ganz leise. Das Gerät wird zur Reparatur gebracht. Der Reparaturdienst meint: Da ist nichts kaputt, sie haben vermutlich nur etwas falsch gemacht. Tatsächlich: Neben dem Ein-/Ausschalter liegt ein Knopf zur Modus-Umschaltung. Auf den kann man versehentlich kommen, wenn man das Gerät ein- oder ausschaltet. Später hatte ich einmal einen echten Reparaturfall mit dem Gerät. Die Dame an der Reparaturannahme fragte gleich: „Haben sie vielleicht versehentlich den Modus umgeschaltet?" - Der Fall kommt demnach öfter vor.

Telefonnummern speichern. Das Einspeichern der Nummern in den Telefonapparat ist eine aufwendige Prozedur. Das Dumme ist, dass es keine Rückmeldung gibt, inwieweit das Vorhaben geglückt ist. Man muss einen Probeanruf machen und es nötigenfalls erneut versuchen.

Textverarbeitung. Ich bringe einen Teil des Textes aus diesem Kapitel nach hinten, in die Datei des 14. Kapitels. Anschließend arbeite ich weiter und vergesse diesen Transport. Am Ende der Sitzung kommen die üblichen Fragen: Änderungen speichern? ja/nein/abbrechen. Bei der Datei, die ich gerade bearbeite, lasse ich speichern. Bei den anderen ist mir das Abspeichern suspekt und ich verneine. Damit habe ich die Arbeit mehrerer Stunden versehentlich vernichtet.

Fall 6: Verfehlte Automatisierung. Im Textverarbeitungsprogramm Word sind in der Grundeinstellung „Komfortfunktionen" eingeschaltet, die eher schaden als nützen: Ich konnte meinen Namen nie auf Anhieb richtig eingeben, weil das Autokorrekturprogramm aus „Timm" stets „Team" machte. Die Ersetzung des Wertes „1/2" durch das Symbol „½" wird meist nicht als ärgerlich empfunden. Der Ärger kommt erst, wenn man das Schriftstück elektronisch übermittelt und das Programm des Empfängers mit diesem Sonderzeichen nichts anfangen kann und stattdessen einen einfachen Unterstrich ausgibt.

Ursachenanalyse: Routinetätigkeiten sollte man den Maschinen überlassen. Aber oft führt die Automatisierung nicht zur gewünschten Entlastung des Menschen. Das System wird komplizierter und es erfordert zusätzliche Kontrollen. Und dabei kommt es zu Wahrnehmungs- und Aufmerksamkeitsfehlern.

Fall 7: Grafische Benutzeroberflächen. Im Menü der Bedienoberfläche von Delphi findet man den Aufruf der „Ansicht überwachter Ausdrücke" im Menü Ansicht. Aber wie lassen sich neue Ausdrücke hinzufügen? Nach längerer Suche findet man den Befehl „Ausdruck hinzufügen" im Menü Start. Ähnliches hat das Textverarbeitungsprogramm Word zu bieten: „Sortieren" findet man im Menü Tabellen! Häufiger kommt (zumindest bei mir) das Sortieren von nicht tabellarisch erfassten Informationen vor.

Ursachenanalyse: Gibt es bei den Funktionselementen eine räumliche oder logische Gruppierung, sollte die sich in ähnlicher Weise in den Bedienelementen wiederfinden. Als weiteres natürliches Prinzip gilt, dass jedes Bedienelement möglichst in der Nähe des Funktionselements liegt. Wie hier wird auch in den folgenden Fällen gegen natürliche Abbildungs- und Ordnungsprinzipien verstoßen.

Lichtschalter. Im Esszimmer in einer Garderobennische befinden sich zwei Schalter, einer ist für das Licht in der Garderobe. Darüber ist der Schalter für das entferntere Licht im Esszimmer

angebracht. An diese Anordnung kann man sich nicht gewöhnen. Noch nach Jahren wird immer mal wieder der falsche Schalter betätigt.

Fenster schließen. Der Knopf [×] zum Schließen eines Fensters bei Windows ist an hervorragender Stelle, nämlich äußerst rechts oben, angeordnet. Diese Position ist so verlockend, dass man oft versehentlich ein Fenster schließt, anstatt es nur zu minimieren.

Gangschaltung. Die beiden Hebel der Gangschaltung sind an der Lenkstange des Fahrrads links und rechts angeordnet. Auf der einen Seite kommt der höhere Gang, wenn man den Schalthebel herunterdrückt, auf der anderen ist es umgekehrt. Die Verwechslung ist programmiert.

Weiterer Design-Plunder: Viele Beispiele für misslungenes Design von Alltagsgegenständen und Hinweise darauf, wie man es besser machen kann, enthält der Klassiker „The Design of Everyday Things“ von Donald A. Norman (1988). Ein geradezu ideales Studienobjekt für Design-Fehler sind die heutigen Telefone. Norman hat dazu einiges zu sagen.

Telefone. Die frühen Telefone entwickelten sich langsam, über mehrere Generationen. Die Geräte wurden schwer und robust. Ließ man eins fallen, dann ging es nicht kaputt; sogar die Verbindung blieb meist erhalten. Die Apparate waren mit einfachen Rückmeldesignalen ausgestattet. Man wusste immer, woran man war. Nun haben wir den Wettbewerb und das wilde Verlangen, möglichst schnell neue Produkte mit möglichst viel modischem Schnickschnack auf den Markt zu bringen. Dabei gehen manche nützlichen Eigenschaften verloren: Viele Telefone geben keine Rückmeldung, wenn die Tasten gedrückt werden. Manche sind so mit Funktionen überladen, dass man oft die einfachsten nicht mehr finden kann. Einige Geräte sind so leicht, dass sie beim Gespräch öfter einmal herunterfallen. Zu allem Überfluss ist der Hörer mit dem Gerät über ein Spiralkabel verbunden.

Es gibt Design-Fehler, die sich in kein Schema einordnen und aus denen sich keine besonderen Lehren ziehen lassen, weil es sich allzu offensichtlich um Plunder handelt. Diese Fehler können höchstens zur Erheiterung beitragen. Hier ist ein solcher.

Designer Computer. Das Äußere des Computers stammt von einem weltberühmten Designer. Wenn man die Tastatur des Computers nach getaner Arbeit erlöst von sich wegschiebt, kommt es zum Neustart. Nicht gespeicherte Arbeitsergebnisse gehen verloren. Der Reset-Knopf des elegant gestylten Geräts liegt genau auf Höhe der Tastatur.

13.5 Regeln für die Entwicklung bedienbarer Maschinen

Ein Bedienfehler impliziert, dass der Mensch, gemessen an den Eigenschaften der Maschine, etwas falsch macht. Der Mensch kann selbst etwas gegen diese Fehler tun - beispielsweise Warnzeichen besser beachten. Dazu muss er deren Ursachen auf den Grund gehen können. Das heißt aber auch, dass der Bediener etwas von der Maschine verstehen muss, wenn er Warnzeichen richtig deuten will. Und schon sind wir doch wieder bei den Maschinen: Sie müssen so gebaut sein, dass der Bediener und Anwender sie verstehen kann. Sie müssen dem Menschen mit seinen angeborenen oder erworbenen Fähigkeiten und Schwächen entgegenkommen.

Deshalb sollen hier nicht in erster Linie Regeln abgeleitet werden, die der Bediener beherzigen muss. Diese sind wohl bereits bei den Ursachenanalysen des letzten Abschnitts hinreichend klar geworden. Jetzt wenden wir uns der anderen Seite des Mensch-Maschine-Systems zu: Wir leiten aus den beobachteten Bedienfehlern und den Ursachenanalysen Regeln für das Design besser bedienbarer Maschinen ab.

Im Fall 1 der Fallsammlung sind es die vielen Fehlermeldungen und Warnungen, die manchmal schwere, manchmal leichte und manchmal sogar nur vermutete Mängel signalisieren und mit denen der Bediener nicht klarkommt. Diese Flut muss eingedämmt werden. Aber wie?

Die Flut teilweise irrelevanter Meldungen lässt sich hier auf die zugrundeliegende Programmiersprache C++ (Stroustrup, 1991) zurückführen - ein Sprachungetüm, das an seiner schieren Größe krankt. Die Sprache hat viele teilweise einander funktionell überlappende Eigenschaften und Ausdrucksmöglichkeiten, dass es kaum möglich ist, eine Entwicklungsumgebung zu bauen, die jeweils nur die gerade relevanten Meldungen ausgibt. Kurz: Die Programmiersprache und die darauf aufbauenden Systeme sind zu kompliziert und durch Entwickler oder Anwender kaum beherrschbar. Das ist die Hauptquelle des Übels.

Die Kompliziertheit liegt auch dem Bedienfehler des Falles 5 (Typing fast) zugrunde. Es ist die Ironie der „fetten" Systeme, dass sie einerseits dem Anwender immer mehr Funktionen und immer bequemere Zugangsmöglichkeiten zu diesen Funktionen bieten wollen, und dass sie andererseits gerade dadurch immer schwerer bedienbar werden. Hätte das System weniger Möglichkeiten der Modeumschaltung geboten, wäre der Bedienfehler wohl nicht passiert.

Aus diesen Analysen ergibt sich die erste und vordringliche Forderung an den Konstrukteur bedienerfreundlicher Maschinen, nämlich die

Regel 1. Strebe nach *Einfachheit.*

Einfach ist ein System, wenn seine Funktion leicht *überprüfbar* und *verständlich* ist. Das einfache System enthält nur die notwendigsten der geforderten Eigenschaften und Fähigkeiten und ein *Minimum an Komponenten,* und diese sind möglichst *schwach miteinander verkoppelt* (Popper, 1982; Wirth, 1994; Leveson, 1995). Kurz: es ist ein schlankes System.

Es gibt aber auch Ausnahmen von der Regel: Das moderne Verkehrsflugzeug ist ein hochkomplexes stark verkoppeltes System mit großem Gefährdungspotential. Dennoch ist es eins der sichersten Verkehrsmittel. Aber gerade hier werden die größten Anstrengungen für die richtige Aufgabenaufteilung zwischen Mensch und Maschine unternommen (Perrow, 1995). Siehe dazu die Erläuterungen zur Regel 5.

Regel 2. Sorge für *Direktheit* der Manipulation und Kommunikation. Schaffe eine *selbsterklärende Bedienoberfläche*. Mache die Funktion *sichtbar*.

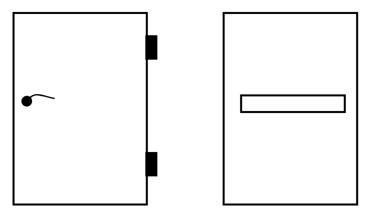


Bild 13.5 Klassische Tür (links) und moderne Tür (rechts)

Die Bedienoberfläche sollte Signale geben, so dass ein adäquates mentales Modell des Artefakts induziert und die korrekten Bedienhandlungen angeregt werden. Ein gutes Beispiel ist „die klassische Tür": Aufgesetztes Türblatt und Anbringung der Klinke sagen, was zu tun ist. Die „moderne Tür" hingegen verheimlicht ihre Funktion (Bild 13.5). Das schlechte Design kommt deswegen meist mit einer Bedienungsanleitung daher. Es wird eigens draufgeschrieben, ob man ziehen oder drücken muss. Solche Erläuterungen stellen kommunikative Umwege dar; sie widersprechen dem Prinzip der Direktheit der Manipulation und Kommunikation. Die Sache mit dem Wasserhahn (Fall 2) ist ein weiteres Beispiel für die Missachtung des Prinzips der selbsterklärenden Bedienoberfläche.

Das Design des eben erwähnten Wasserhahns verstößt - wie auch das des Fotoapparats aus Fall 3 - gegen die

Regel 3. *Bedenke die Gewohnheiten* des Bedieners und Anwenders, nutze seine *Routine* und beachte *Konventionen*.

Gegen diese Regel hat beispielsweise der Designer der Web-Page (Fall 4) verstoßen, indem er die Stichwörter über die Fläche verstreute. Um der Leseroutine entgegenzukommen, sollten Stichwortlisten auf Buchseiten, Web-Pages oder Bedienoberflächen eines Computerprogramms grundsätzlich linksbündig ausgerichtet werden.

Zu den Konventionen gehören die *allgemein akzeptierten Codes* für Darstellungs- und Bedienungselemente. Beispielsweise steht im „Farbcode“

- Rot für halt, heiß oder Gefahr,
- Grün für weiter oder sicher,
- Gelb für Vorsicht und
- Blau für kalt.

Wenn Maschinen den Menschen verunsichern, indem sie ihn darüber im Unklaren lassen, was gerade passiert (Fall 5), dann ist das ein Verstoß gegen die

Regel 4. Sorge für die *Rückmeldung* des Erfolgs oder Misserfolgs von Aktionen und für Informationen über den Maschinenzustand.

Das einleitende Beispiel dieses Kapitels - der Redakteur vor seinem Schreibsystem mit Korrekturprogramm - wirft die Frage auf, inwieweit die Entlastung des Bedieners und Anwenders durch Automatisierung überhaupt hilfreich ist. Das Korrekturprogramm verlangt vom Redakteur zusätzlichen Kontrollaufwand. Von einer geradezu verfehlten Automatisierung wird im Fall 6 berichtet. Solche Beobachtungen sind der Grund für die

Regel 5. Achte auf eine zweckmäßige *Aufteilung der Funktionen* auf Mensch und Maschine unter Berücksichtigung der jeweiligen Fähigkeitsprofile.

Die Regel verlangt, das zu automatisieren, was die Maschine besser kann als der Mensch und auf Automatisierung dort zu verzichten, wo dem Menschen durch Automatisierung Tätigkeiten aufgebürdet werden, in denen er nicht gut ist.

Einen ersten Anhaltspunkt für eine solche Aufteilung bietet die Liste der Fähigkeitsprofile von Fitts (Kasten „Liste von Fitts“). Sie ist zu ergänzen, insbesondere um den Hinweis, dass der *Mensch beim Überwachen nicht besonders gut* ist (Sheridan, 2000; Bubb, 1997). Eine weitgehende Automatisierung drängt den Menschen aber gerade in die Rolle des Überwachers.

Von Lisanne Bainbridge stammt der Begriff von der *Ironie der Automatisierung* (Reason, 1994, S. 224): Ein weitgehend automatisiertes System enthält dem Bediener die Gelegenheiten zum Einüben der im Ernstfall wichtigen Fertigkeiten vor. Letztlich wird der Gewinn, den die Automatisierung verspricht, durch das zusätzlich erforderliche Operateur-Training teilweise wieder aufgefressen.

Nancy Leveson (1995, S. 118) stellt folgenden Zusammenhang her: Wenn vom Operateur nur geringe Mitwirkung gefordert wird, kann das zu einer geringeren Wachsamkeit und zu einem übermäßigen Vertrauen in das automatisierte System führen. Und Mica R. Endsley (1995) stößt ins gleiche Horn, wenn sie konstatiert, dass das Situationsbewusstsein verloren gehen kann, wenn die Rolle des Bedieners auf passive Kontrollfunktionen reduziert wird.

Felix von Cube (1995, S. 75) sieht die Langeweile als Ursache des Übels: „Dadurch, dass der Unterforderte seine Aufmerksamkeit nicht oder nur zum geringen Teil für seine Arbeit einzusetzen braucht, richtet er sie auf andere Bereiche. So wird sie unter Umständen ganz von der Arbeit abgezogen, es kommt zu gefährlichen Situationen.“

Der richtigen Funktionsaufteilung gilt das Hauptaugenmerk bei der Cockpitgestaltung von Flugzeugen. Entsprechend gut ist dieses Thema untersucht (Sharit, 1997; Sheridan, 1997).

Liste von Fitts (Sharit, 1997)

Menschen sind heutigen Maschinen überlegen hinsichtlich der Fähigkeiten

- geringe Mengen visueller oder akustischer Energie aufzuspüren,
- optische oder akustische Muster wahrzunehmen,
- zu improvisieren und flexibel zu reagieren,
- große Mengen Information über lange Zeit zu speichern und sich zur passenden Zeit an die relevanten Fakten zu erinnern,
- der Induktion, also Erweiterungsschlüsse zu machen,
- Entscheidungen zu fällen.

Heutige Maschinen sind Menschen überlegen hinsichtlich der Fähigkeiten

- schnell auf Steuerungssignale zu reagieren und große Kraft präzise einzusetzen,
- Routineaufgaben durchzuführen,
- Information kurzfristig zu speichern und auf Aufforderung hin vollständig zu löschen,
- logische und arithmetische Kalkulationen durchzuführen,
- hochgradig komplexe Operationen durchzuführen, das heißt, viele verschiedene Dinge auf einmal zu tun.

Der Fall 7 belegt die Notwendigkeit der

Regel 6. Nutze die natürlichen Ordnungsprinzipien und die Gestaltgesetze, insbesondere das Gesetz der Gruppierung und der Nähe. Ordne die Funktionselemente und deren Bedienungselemente abbildungstreu einander zu.

Wenn das alles nicht ausreicht, oder wenn das Learning by Doing und die Exploration des Systems angeregt werden sollen, tue ein Übriges entsprechend der

Regel 7. Schaffe die verzeihende Bedienoberfläche. Erlaube das Rückgängigmachen von Fehleingaben.

Die Gestaltung der Bedienoberflächen heutiger Computer-Programme ist Gegenstand des Buches von Ben Shneiderman (1998). Speziell im Hinblick auf das sicherheitsgerichtete Design von Mensch-Maschine-Schnittstellen hat Nancy Leveson 60 Regeln zusammengestellt (1995). Einen Überblick und Hinweise auf weiterführende Literatur bietet der Artikel Yili Liu (1997).

14 Konstruieren nach dem Fehlerintoleranz-Prinzip

Schwerpunkt dieses Kapitels bilden die Methoden zur Konstruktion zuverlässiger Software. Dies trägt der Tatsache Rechnung, dass in heutigen komplexen Systemen die Software-Fehler eine herausragende Rolle spielen und dass sie teuer zu stehen kommen. Eine herausragende Rolle spielen sie vor allem deshalb, weil sich die Entwicklungs- und Konstruktionsanstrengungen zunehmend auf die Software verlagern. Etwa seit den achtziger Jahren ist das Versagen komplexer Systeme häufiger auf Software-Fehler als auf Hardware-Fehler zurückzuführen - und das trotz der gewaltigen Fortschritte bei den Software-Entwicklungsmethoden.

Schon seit Beginn des Computerzeitalters schlägt sich die Zunft der Software-Hersteller mit dem Problem herum, dass sich ihr Baumaterial schneller fortentwickelt als die Fähigkeiten und Methoden, damit nützliche Werke zu bauen. Beispiele dafür sind in großer Zahl dokumentiert (Neumann, 1995). Zu den Ursachen der sogenannten *Software-Krise* zählen die folgenden (Balzert, 1996, S. 25 ff.):

- Software bietet einen großen Gestaltungsspielraum.
- Die realisierbaren Funktionen sind nicht durch physikalische Gesetze begrenzt.
- Software ist leichter und schneller änderbar als Hardware.

Bereits im Jahr 1969 prägte der Münchner Informatik-Professor Friedrich L. Bauer auf einer NATO-Wissenschaftstagung in Garmisch das Wort *Software Engineering*. Dahinter stecken der Anspruch und die Hoffnung, die Software-Entwicklung ingenieurmäßig zu betreiben und dadurch der Probleme Herr zu werden.

Im Laufe der letzten fünfzig Jahre tauchten viele Vorschläge für bessere Entwicklungsmethoden auf. Zwei Richtungen lassen sich deutlich unterscheiden. Die Vertreter der einen Richtung finden sich damit ab, dass Entwicklungs- und Programmierfehler nun einmal passieren und sie konzentrieren sich auf Techniken, mit denen sich die Fehler kontrollieren lassen. Zu diesen Techniken gehören die

- Zuverlässigkeitswachstumsmodelle zu Prognosezwecken,
- Software-Redundanztechniken zur Tolerierung eingebauter Fehler und
- vieles von dem, was heute unter Software-Wartung verstanden wird (Beta-Releases beispielsweise und „Bananen-Software“, die beim Kunden reift).

Fehlertoleranz ist bei Maschinen von Vorteil, wenn damit gemeint ist, dass Hardware-Ausfälle oder Bedienfehler nicht notwendigerweise zu (schwerwiegender) Fehlfunktion führen. Demgegenüber sollten wir Konstruktionsfehler niemals tolerieren. Das Tolerieren der selbst fabrizierten Fehler gehört sicher nicht zu den Tugenden des Ingenieurs.

Wir setzen besser auf Techniken und Methoden, die Konstruktionsfehler möglichst von vornherein verhindern oder doch zumindest eine Entwicklung des Konstruktionsprozesses in Richtung Fehlerfreiheit fördern. Wer so handelt, folgt dem *Fehlerintoleranz-Prinzip*.

Die Verfahren und Methoden nach dem Fehlerintoleranz-Prinzip enthalten intuitive und diskursive Elemente. Wesentliche Merkmale sind

1. das Lernen aus den Fehlern und Erfahrungsfortschreibung,
2. die Programmstrukturierung mit geeigneten Programmiersprachen,
3. das beweisgeleitete Programmieren mit halbformalen und formalen Methoden

4. Tests nach der negativen Methode durch eine von der Programmerstellung unabhängigen Stelle,
5. Code-Inspektionen und Walkthroughs im Team.

Wir konzentrieren uns auf die Punkte 1 bis 3, also auf die eigentliche Konstruktion nach dem Fehlerintoleranz-Prinzip. Hinsichtlich der Fehlerkontrolle durch Tests mit und ohne Computereinsatz (Punkte 4 und 5) ist das Buch von Myers (1987) ein ausgezeichneter Ratgeber.

Manchmal reicht es aus, den Anwendungsrahmen einer Methode und die Ziele, die man damit verfolgt, zu wechseln. Schon wird sie - auch wenn das zunächst gar nicht beabsichtigt war - zu einem Instrument des Fehlerintoleranz-Prinzips. Genau das wird im Kapitel 11 demonstriert, und zwar durch Umdeutung des Zwecks der Zuverlässigkeitswachstumsmodelle: Beim Prognosewerkzeug steht die Fehlertoleranz im Vordergrund. Nutzt man die Zuverlässigkeitswachstumsmodelle retrospektiv, werden sie zu einem Werkzeug im Sinne des Fehlerintoleranz-Prinzips.

Umfassende Darstellungen der Methoden zur Konstruktion fehlerfreier Software findet der Leser beispielsweise in den Büchern von Balzert (1996, 1999). Auf die individuelle Programmiertätigkeit zugeschnitten ist das Buch von Humphrey (1995). Zu den Klassikern des Gebiets gehören das Buch von Linger, Mills und Witt (1979) und die Aufsätze der Sammlung von Dahl, Dijkstra und Hoare (1972).

Bevor wir uns die Methoden zur Konstruktion fehlerfreier Software näher ansehen, fragen wir danach, welche Denkmuster uns dazu bringen, Konstruktionsfehler zu machen.

14.1 Die angeborenen Lehrmeister und ihre dunkle Kehrseite

Einen Aspekt des Falles 1 aus dem letzten Kapitel („Warnungen sind harmlos“) haben wir bisher nicht erörtert: Die Studenten hatten eine Hypothese, woran es wohl gelegen haben mag, dass ihr Programm nicht funktionierte. Sie „beschuldigten“ ein im Grunde harmloses und korrektes Unterprogramm, den Fehler verursacht zu haben. Solche voreiligen „Schuldzuweisungen“ gehören zu den wirkungsvollsten und stabilsten Mechanismen der Zeitverschwendung.

Die Sache läuft fast immer nach einem Schema, das jeder von uns kennt: Ein Fehler wird beobachtet. Ein Verdacht, woran es liegen könnte, kommt einem in den Sinn. Es gibt also eine Hypothese für die Ursache. Wir wissen, dass bei Vorliegen der Hypothese der Fehler mit großer Wahrscheinlichkeit folgt. Vielleicht sind wir gar sicher, dass bei Vorliegen der Hypothese der Fehler auftreten muss.

Da uns weitere mögliche Ursachen nicht ohne Weiteres in den Sinn kommen, ziehen wir den kühnen und ungerechtfertigten Umkehrschluss, dass aufgrund der Beobachtung die Hypothese höchstwahrscheinlich ist.

Wir machen eventuell einige weitere Beobachtungen, die diese Vermutung bestätigen. Im vorliegenden Fall ist nach Austausch des inkriminierten Unterprogramms der Fehler scheinbar weg. Damit wird uns die Hypothese zur Gewissheit.

Das ist voreilig, wie sich meist viel zu spät herausstellt. Es gibt viele Anzeichen dafür, dass wir dazu neigen, Ursachen und Konsequenzen zu verwechseln, und dass wir plausible Schlussfolgerungen vorschnell zur Gewissheit erheben. Diese Fehler gehören zu einer Klasse von *Irrtümern*, die dadurch charakterisiert sind, dass

- sie einem auch in entspannter Atmosphäre und bei guter körperlicher Verfassung passieren;

- höhere kognitive Ebenen, insbesondere das bewusste und problemlösende Denken, beteiligt sind;
- sie von vielen Personen in derselben Art und Weise und fast zwangsläufig begangen werden;
- ihnen durch Kontrollen durch einen selbst oder durch andere schlecht beizukommen ist.

Diesen Irrtümern liegen Denkfallen zu Grunde. Und dabei verwende ich den Begriff Denkfalle in einer ganz bestimmten Bedeutung.

Eine *Denkfalle* ist gegeben, wenn eine Problemsituation einen weit verbreiteten und bewährten Denkmechanismus in Gang setzt, und wenn dieser Denkmechanismus mit der gegebenen Situation nicht zurechtkommt und mit hoher Wahrscheinlichkeit zum Irrtum führt.

Weit verbreitet und bewährt heißt, dass dieser Denkmechanismus zum Hintergrundwissen einer ganzen Gruppe von Personen - beispielsweise der Gemeinde der Programmierer oder der Angehörigen einer Zivilisation - gehört. Folglich werden Fehler aufgrund von Denkfallen in dieser Population mit hoher Wahrscheinlichkeit und immer wieder begangen. Und diese Fehler entziehen sich den üblichen Kontrollmechanismen, wie beispielsweise dem erneuten Durchlesen eines Programmtexts durch dessen Urheber oder durch andere.

Denkfallen können die Ursachen von riskanten Manövern, Fehldiagnosen, Design-, Konstruktions-, Programmier- und Bedienfehlern sein. Sie begegnen dem Designer, Konstrukteur, Fahrer, Piloten, Operateur, also jedem, der technisches Gerät baut oder mit ihm umgehen muss.

Denkfallen gehen offenbar auf an sich nützliche Denkmechanismen zurück. Das kann gar nicht anders sein. Denn wären sie nicht nützlich, könnten sie gar nicht weit verbreitet sein. Die mit ihnen ausgestatteten Populationen hätten im Überlebenskampf keine Chance und wären längst untergegangen.

Eine herausragende Rolle unter diesen Denkmechanismen spielen die *angeborenen Lehrmeister*, wie Konrad Lorenz (1973) sie nennt: „Die angeborenen Lehrmeister sind dasjenige, was vor allem Lernen da ist und da sein muss, um Lernen möglich zu machen“. Bei Rupert Riedl (1981) finden wir sie unter dem Namen Voraus-Urteile wieder: „Das biologische Wissen enthält ein System vernünftiger Hypothesen, Voraus-Urteile, die uns im Rahmen dessen, wofür sie selektiert wurden, wie mit höchster Weisheit lenken; uns aber an dessen Grenzen vollkommen und niederträchtig in die Irre führen“.

Zu den angeborenen Lehrmeistern zähle ich die *Strukturerwartung*, die *Kausalitätserwartung* und die Anlage zur *Induktion*. Die folgenden Abschnitte sollen die Rolle klar machen, die diese Lehrmeister im Laufe des Baus von Systemen spielen und dass sie hinter dem einen oder anderen Fehler stecken.

14.2 Strukturerwartung

Alles Leben geht augenscheinlich von der Hypothese eines objektiv existierenden Kosmos aus, der von Recht und Ordnung zusammengehalten wird. Diese Strukturerwartung hat sich im Laufe der Evolution als Erfolgsrezept erwiesen. Die Strukturerwartung wirkt sich bei der optischen Wahrnehmung als Prägnanztendenz aus - das ist die Sinnsuche des Wahrnehmungsapparats (13. Kapitel). Auch auf höheren kognitiven Ebenen suchen wir stets nach Gesetzmäßigkeiten größtmöglicher Einfachheit und Regularität. Die Einebnung und auch die Übertreibung von Charakteristika sind Vereinfachungen hin zur „guten Gestalt“ (Eibl-Eibesfeldt, 1984, S.64).

Nehmen wir als Beispiel die *mentalen Landkarten*. Wenn ich Kursteilnehmer frage, ob London weiter östlich oder weiter westlich, weiter nördlich oder weiter südlich als Berlin liegt, be-

komme ich mehrheitlich die Antwort: London liegt nordwestlich von Berlin. Dabei liegt London weiter im Süden als Berlin. Woher kommt der Irrtum bezüglich der Nord-Süd-Richtung?

Bei der „mentalen Grenzziehung“ kommt es aufgrund der Prägnanztendenz zu Begradigungen und Vergröberungen. An diesen prägnanten Grenzen orientieren wir uns. Folglich werden Orte in Gedanken systematisch verrückt (Anderson, 1988, S. 94). Im Fall London-Berlin erkläre ich mir die zu beobachtende Verzerrung so: Bezogen auf die Nordsee liegt England im Westen und Deutschland im Süden. Der Ärmelkanal wird auf seine Ost-West-Richtung hin „begradigt“. Durch diese Vereinfachung „rutscht“ England gedanklich gen Norden.

Hier noch ein Beispiel aus der Programmierung:

Formeln lax. Der mathematische Ausdruck $a < x < b$ wird üblicherweise gelesen als $(a < x) \wedge (x < b)$. Aber der Computer denkt nicht daran, das zu tun – er denkt nämlich überhaupt nicht. Schlimm ist, wenn der Computer einen solchen Ausdruck nicht bereits beim Compilieren ablehnt, sondern stattdessen uminterpretiert. Gerade in der Programmiersprache C passiert dies. Die Bedingung „660 <= ArrivalTime < 840“ ist stets erfüllt, egal welchen Wert ArrivalTime hat. Den Ausdruck „x<=y<z“ arbeitet der C-Compiler nämlich von links nach rechts ab, also wie „(x<=y)<z“; und der geklammerte Ausdruck ist gleich 0 oder gleich 1, jedenfalls also kleiner als 840. Das Programmierteam hatte sich das ganz anders vorgestellt. Aufgrund der Strukturerwartung können wir die Formel richtig interpretieren. Wir lesen in die im Grunde unsinnige Formulierung etwas Passendes hinein. Der Computer kann das nicht.

Wir sind ständig bestrebt, Ordnung in der Welt zu finden. Und wenn eine solche nicht von vornherein gegeben ist, bringen wir durch Klassifizierung und Kategorisierung Ordnung hinein. Der Prägnanzdruck verleitet uns dazu, in Gegensätzen zu denken, er führt zur Polarisierung des Denkens. „Vorhandene Gegensätze werden nach dem *Prinzip der Kontrastbetonung* hervorgehoben, und das vermittelt Orientiertheit und Klarheit, führt aber andererseits zu einer Vereinfachung des Weltbildes“ (Eibl-Eibesfeldt, 1984, S. 138; Hervorhebung von mir).

Wir teilen Aufgaben in wichtige und weniger wichtige ein. Das hilft uns, die zur Verfügung stehenden Ressourcen möglichst wirkungsvoll einzusetzen. Es führt aber auch nahezu zwangsläufig dazu, die *vermeintliche Nebensachen* in ihrer Bedeutung zu unterschätzen. Auch hierfür ist der Fall 1 des letzten Kapitels ein gutes Beispiel. Warnungen wurden von den Programmierern - anders als die „echten“ Fehlermeldungen - als harmlos eingestuft.

Die Anforderungsklassen und Risikobereiche (Kapitel 6) stellen ein Musterbeispiel für ein Klassifizierungsschema dar. Diese Klassifizierung bewirkt eine Konzentration der Aufmerksamkeit auf die als sicherheitsrelevant eingestuften Systeme. Den geringer geachteten Systemen wird oft zu spät größere Aufmerksamkeit zugewendet, und zwar erst bei der Analyse eines Unfalls, zu dem sie ihren Beitrag geleistet haben.

Three Mile Island (1): Die für Genehmigung und Überwachung der Kernkraftwerke der USA zuständige Atombehörde NRC verlangte - zumindest in der Zeit vor dem Reaktorunfall von Three Mile Island (28.3.1979) -, dass Qualitätssicherungsprogramme auf sicherheitsrelevante Systeme anzuwenden sind. Für andere Teile eines Kraftwerks bestand diese Forderung nicht. Tatsächlich zählten aber einige Fehler in nicht als sicherheitsrelevant eingestuften Komponenten zu den Ursachen des Unfalls von Harrisburg. Dazu gehörte der Defekt eines Abblaseventils. Dieses Ventil schloss im entscheidenden Moment nicht. Die Operateure wurden dessen nicht gewahr, da es keine direkte Stellungsanzeige für dieses Ventil gab (Lewis, 1980; Leveson, 1995).

Der gescheiterte Jungfernflug der Ariane 5: Am 4. Juni 1996 scheiterte der Jungfernflug der europäischen Trägerrakete Ariane 5. Auslösendes Ereignis war die Bereichsüberschreitung einer Variablen innerhalb der Software an Bord der Rakete etwa 30 Sekunden nach dem Start.

Das fehlerauslösende Software-Modul war praktisch unverändert von der Ariane 4 übernommen worden. Details (Lions, 1996):

1. Vier der sieben Variablen im fehlerauslösenden Software-Modul waren gegen Bereichsüberschreitungen geschützt. Die fehlerauslösende Variable war eine von drei Variablen, bei denen man auf diesen Schutz verzichtet hatte, weil aufgrund der Flugbahn der Ariane 4 mit Bereichsüberschreitungen nicht zu rechnen war. Die Flugbahn der Ariane 5 ist von derjenigen der Ariane 4 verschieden. Es kam zur Bereichsüberschreitung. Das fehlerauslösende Modul wird bei der Ariane 5 nach dem Start eigentlich gar nicht benötigt. Dass es dennoch aktiv war, geht auf Software-Anforderungen für die Ariane 4 zurück.
2. Das fehlerauslösende Software-Modul ist Bestandteil eines Gerätes, das aus zwei hard- und softwaremäßig identischen Subsystemen besteht. Das Redundanzkonzept soll Fehlertoleranz gegenüber Hardware-Ausfällen bewirken. Kommt es zu einer Ausnahmesituation in der Software, dann gibt das betroffene Subsystem eine Fehleranzeige auf die Datenleitungen und schaltet sich ab. Das vorübergehende Versagen wird so in einen Ausfall des Geräts transformiert.
3. Wegen der identischen Software fielen beide Subsysteme quasi gleichzeitig aus, so dass der On-Board-Computer nicht auf das parallele Subsystem umschalten konnte.

Die anstelle der Flugdaten interpretierten Diagnosedaten bewirkten eine Maximalauslenkung der Steuerdüsen. Das führte zu einem großen Auslenkwinkel der Rakete von 20° gegenüber der Flugrichtung. Unter der hohen Windbelastung zerbrach die Rakete. Der Selbstzerstörungsprozess wurde ausgelöst.

Analyse: Ein Glaubenssatz der Ingenieure, nämlich „Change is bad“ hat zur Beibehaltung eines bewährten Designs unter veränderten Bedingungen geführt (Punkt 1). Die *Blickverengung* und die Denkfalle *Bewährt aber falsch* taten ein Übriges. Punkt 2 ist Beispiel für übertrieben einfache Klassifizierungen und abgestufte Qualität: Es wird nur zwischen korrektem Verhalten einerseits und den Abweichungen von der Norm andererseits unterschieden. Versagen, Ausnahme und Ausfall landen in derselben Kategorie, ohne zu bedenken, dass sie unterschiedliche Abhilfemaßnahmen erfordern. Und Punkt 3 schließlich geht auf den *Glauben an die Redundanz* zurück.

14.3 Kausalitätserwartung

Kausalitätserwartung ist die „Erwartung, dass Gleiches dieselbe Ursache haben werde. Dies ist zunächst nicht mehr als ein Urteil im Voraus. Aber dieses Vorurteil bewährt sich ... in einem derartigen Übermaß an Fällen, dass es jedem im Prinzip andersartigen Urteil oder dem Urteils-Verzicht überlegen ist“ (Riedl, 1981). Verhängnisvoll wird das Prinzip bei ausschließlich linearem Ursache-Wirkungs-Denken, und wenn wir die Vernetzung der Ursachen und die Nebenwirkungen unserer Handlungen außer Acht lassen (Dörner, 1989). Die vom Menschen verursachten Umweltprobleme zeugen davon.

Noch ein Beispiel: Meist wird nach einem Flugzeug-, Bahn-, Schiffsunglück oder einem Kernkraftwerksunfall der Pilot, der Lokführer, der Kapitän oder der Operateur als Schuldiger präsentiert. Menschliches Versagen heißt es dann, obwohl man besser von einer Fehlanpassung von Mensch und Aufgabe (bzw. Maschine) reden und von einer Vielzahl von Ursachen ausgehen sollte.

Die Fehlerbaumanalyse kann helfen, das monokausale Denken zu überwinden und die meist vielgestaltigen Ursachen unerwünschter Ereignisse mit schädlichen oder gar katastrophalen Folgen zu erfassen (Leveson, 1995).

Die Kausalitätserwartung verhindert manchmal nicht nur das Auffinden der wahren Ursache, weil wir meinen, die Ursache bereits gefunden zu haben; sie bewirkt auch, dass wir Ursachen sehen, wo gar keine zu finden sind. Dies dürfte wohl eine der häufigsten Ursachen der Fehlinterpretation von Statistiken sein. Die folgende Statistik demonstriert diesen Effekt.

Xenophobie. Im Städtchen *Falldala* mit 20 000 Einwohnern beträgt der Ausländeranteil 30 %. Die folgende Tabelle gibt die Kriminalitätsstatistik wieder. Daraus lässt sich folgern: Ausländer neigen stärker zur Kriminalität als Inländer.

Kriminalitätsstatistik von Falldala

	Einwohner	Straftaten je Jahr	bezogen auf je 1000 Einwohner
Ausländer	6000	51	8.5
Inländer	14000	59	4.2

Bei eingehender Betrachtung der Kriminalitätsstatistik erweist sich, dass dieser Kausalzusammenhang tatsächlich nicht besteht. In einem Stadtteil (nennen wir ihn Aschental) ist die Kriminalität nämlich besonders hoch ist. Nun sind, aufgrund der für Inländer unattraktiven Bebauung aus den 50-er Jahren, vorwiegend Ausländer im Aschental angesiedelt. 5 000 der insgesamt 10 000 Bewohner sind Ausländer. Die übrigen leben in der Innenstadt. Für das Aschental weist die Kriminalitätsstatistik eine ziemlich hohe Kriminalitätsrate aus.

Kriminalitätsstatistik des Stadtteils Aschental

	Einwohner	Straftaten je Jahr	bezogen auf je 1000 Einwohner
Ausländer	5000	50	10
Inländer	5000	50	10

Günstiger sieht es in der Innenstadt von Falldala aus. Die Kriminalitätsrate ist recht niedrig. Insgesamt erweist sich, dass die Ausländer nicht häufiger oder weniger häufig zu Straftaten neigen als Inländer. Die anfängliche Vermutung zur Ausländerkriminalität entpuppt sich als oberflächlich. Die genauere Betrachtung legt ganz andere Deutungen nahe: Vielleicht liegt es an der Umwelt, an der Armut, am Milieu des Stadtteils. Jedenfalls sind tieferliegende Analysen erforderlich.

Kriminalitätsstatistik der Innenstadt von Falldala

	Einwohner	Straftaten je Jahr	bezogen auf je 1000 Einwohner
Ausländer	1000	1	1
Inländer	9000	9	1

Analyse: Durch das unspezifische Aggregieren von Zähldaten werden „Zusammenhänge“ erzeugt, die in den Einzeldaten nicht existieren. Hier ist es ein Zusammenhang zwischen den Attributen „Ausländer“ und „Neigung zur Kriminalität“. Und das Kausaldenken verstärkt diesen Scheinzusammenhang zu einer Kausalbeziehung, die tatsächlich nicht gegeben ist.

Es folgen noch ein paar Beispiele aus der Programmierung.

Shift. Das C-Programm Shift.c wird vom Compiler klaglos übersetzt. Es funktioniert - zumindest in der Programmierumgebung - einwandfrei. Überraschenderweise erfolgt beim Aufruf auf Kommandozeilenebene keinerlei merkbare Reaktion. Das Programm scheint doch

fehlerbehaftet zu sein. Also wird der Fehler mit einiger Ausdauer im Programm gesucht. Sehr spät kommt der richtige Einfall: Auf Kommandozeilenebene gibt es den Shift-Befehl bereits. Genau dieser wird mit dem Kommando Shift aufgerufen. Der Befehlsaufruf verdeckt den Programmaufruf. Einer frühzeitigen Entdeckung dieses Sachverhalts stehen die voreilig gefasste Hypothese („Das Programm muss fehlerhaft sein") und die Blickverengung im Weg.

Irreführende Testdaten. Zur Überprüfung eines Programms zur statistischen Parameterschätzung wurden deterministische Testdaten erzeugt und eingegeben, so dass das Programm die Parameter der Daten exakt hätte reproduzieren müssen. Der Algorithmus und die Computerarithmetik hätten Ergebnisse erwarten lassen, die auf wenigstens 15 Stellen genau sind. Das Ergebnis war aber nur auf fünf Stellen genau. Die Suche nach einem Programmierfehler blieb lange erfolglos. Endlich wurde die Ursache bei den Testdaten entdeckt: Sie wurden nur auf eine Genauigkeit von vier Stellen erzeugt. Das Programm konnte also gar keine besseren Ergebnisse liefern. Warum nun wurde der Fehler so spät entdeckt? Bei einem Test zielt man auf Fehler im Programm. Die naheliegenden Hypothesen für Fehlerursachen betreffen das Programm, und nicht die Testdaten. Der Fehlerfindungsprozess wird durch die gewohnten Induktionsschlüsse und durch lineares Kausaldenken behindert.

Orthogonal-Aufgabe: In Seminaren wurde die Aufgabe gestellt, die Funktionsprozedur Orthogonal gemäß Spezifikation zu programmieren (siehe Abschnitt 1.6). Viele Lösungsvorschläge enthielten Fehler. Eine kleine Kollektion fehlerhafter Antworten:

```
Orthogonal:= FALSE;
IF Alpha-Beta=90 THEN Orthogonal:= TRUE;
```

```
h:= abs(Alpha)-abs(Beta);
IF (h=90) OR (h=270) THEN Orthogonal:= TRUE
ELSE Orthogonal:= FALSE;
```

```
IF abs(Alpha-Beta)=90 THEN Orthogonal:= TRUE
ELSE Orthogonal:= FALSE;
```

```
IF ((Alpha-Beta)<>90) OR ((Alpha-Beta)<>270) THEN
Orthogonal:= FALSE ELSE Orthogonal:= TRUE;
```

Diese Lösungsvorschläge versagen alle beim Testfall $\alpha = 30$, $\beta = 300$.

Die Programme wurden offensichtlich vorwiegend intuitiv geschrieben. Dabei stellt sich der Programmierer die eine oder andere Situation vor, in der Rechtwinkligkeit gegeben ist. Diese Situationen scheinen repräsentativ zu sein. Dieser Eindruck führt zur Blickverengung, welche die Menge der möglichen „Ursachen" schrumpfen lässt. Dem Programmierer kommt hier das eindimensionale Ursache-Wirkungs-Denken in die Quere.

Intuitives Vorgehen beim Entwurf wird grundsätzlich durch Denkfallen bedroht. Wer korrekte Systeme konstruieren will, darf sich nicht allein auf die Intuition verlassen. Weiter unten wird gezeigt, wie sich durch die diskursive Methode solche Fehler vermeiden lassen.

14.4 Die Anlage zur Induktion und das plausible Schließen

Voreilige Hypothesenbildung und die Suche nach bestätigenden Informationen geht auf unsere Fähigkeit und Neigung zur *Induktion* zurück: Wir schließen vom Besonderen (der Beobachtung) aufs Allgemeine (die Hypothese). Alle Theoriebildung geht letztlich auf Induktion zurück. Das macht sie uns unentbehrlich. Aber das ist nur die eine Seite der Sache. Sie hat die bereits angesprochene hässliche Kehrseite.

Fehlerhafte Induktionsschlüsse lassen sich grundsätzlich nicht vermeiden. Karl R. Popper geht sogar so weit, Induktionsschlüsse grundsätzlich als fehlerhaft anzusehen. Jeder auch nur halb-

wegs interessanten Theorie misst er die Korrektheitswahrscheinlichkeit null zu. Auch wenn er zugibt, dass wir mit solchen Theorien oft doch recht weit kommen.

Von *Induktionsfehlern* will ich erst dann sprechen, wenn die Induktion logisch und mathematisch fehlerhaft angewendet wird. Wenn also der allenfalls plausible Schluss auf die Gültigkeit der Hypothese als zwingend überinterpretiert wird, oder wenn die Glaubwürdigkeit der Schlussfolgerung stark überschätzt wird.

Unsere Anlage zur Induktion, also unser Hang zu Erweiterungsschlüssen, arbeitet nach folgendem Argumentationsmuster: Wenn sich aus der Theorie (Hypothese) H ein Ereignis E vorhersagen lässt, und wenn gleichzeitig das Ereignis E aufgrund des bisherigen Wissens recht unwahrscheinlich ist, dann wird die Theorie H aufgrund einer Beobachtung des Ereignisses E glaubwürdiger. Kurz: Aus „H impliziert E“ und „E ist wahr“ folgt „H wird glaubwürdiger“. Diese Art des plausiblen Schließens zusammen mit dem linearen Ursache-Wirkungs-Denken (Kausalitätserwartung) macht generalisierende Aussagen überhaupt erst möglich. So kommen wir zu wissenschaftlichen Hypothesen und schließlich Theorien.

Plausibles statt logisches Schließen: Wir tendieren dazu, Induktionsschlüsse mit größerer Bestimmtheit anzureichern und wie logische Schlussfolgerungen zu interpretieren. Wir unterscheiden nicht konsequent genug zwischen „Aus H folgt E“ und „Aus E folgt H“. Wir wissen: Wenn es regnet, wird die Straße nass. Nun beobachten wir, dass die Straße nass ist, und wir folgern, dass es wohl geregnet haben wird. Dieser Schluss ist zwar plausibel, aber er ist nicht zwingend. Es kann ja auch sein, dass kürzlich der Sprengwagen der Straßenreinigung durchgefahren ist. Dagegen lässt eine trockene (also: nicht nasse) Straße den Rückschluss zu, dass es gerade nicht regnet. Letztere Schlussweise nennt man den Modus Tollens. Der zuvor gezeigt Fehlschluss geht auf das Konto der Denkfalle *Scheitern am Modus Tollens* (Anderson, 1988).

Fehler bei der Hypothesenbildung und -abschätzung: Haben wir eine halbwegs schlüssige Hypothese über die möglichen Ursachen unserer Beobachtungen gefunden, neigen wir dazu, diese Hypothese als einzig mögliche Erklärung der beobachteten Effekte anzusehen und die Suche nach konkurrierenden Hypothesen einzustellen. Voreilige Hypothesen und Ad-hoc-Theorien entfalten eine gewisse Beharrlichkeit: Erst einmal gefasst, geben wir sie ungern auf. Dies bereitet uns Schwierigkeiten, beispielsweise wenn wir eine Diagnoseaufgabe vor uns haben und voreilige Annahmen über die Fehlermechanismen das Aufdecken der eigentlichen Fehlerursache verhindern. Es hat sich herausgestellt, dass diagnoseunterstützende Systeme - in Kraftwerksleitwarten beispielsweise - durch optische Reize das Einfrieren von hinderlichen Vor-Urteilen begünstigen können (Elzer/Kluwe/Boussoffara, 2000, S. 87). Die Untersuchung dieser Fehlermechanismen mittels psychologischer Experimente ist besonders schwer, weil bereits die Experimentatoren in die Denkfalle tappen können und Gefahr laufen, die Versuchsszenarios zu eng zu fassen (Grams, 2000). Ein beliebtes Mittel zur Schaffung und Untermauerung von Vorurteilen sind Statistiken, wie wir am Xenophobie-Beispiel sehen konnten.

Da wir Sicherheit suchen, drängen wir auf die Bestätigung unserer Vorurteile, und weniger auf deren Widerlegung. Damit einher geht unsere Neigung zur *Überbewertung bestätigender Information* (confirmation bias).

Three Mile Island (2). Ein geringer Kühlmittelverlust wurde - entgegen der Vorschrift - von der Bedienmannschaft über lange Zeit akzeptiert. Der Kühlmittelverlust machte sich durch erhöhte Temperatur im Ablaufrohr bemerkbar. Der Kühlmittelverlust an sich war harmlos. Eine bedeutende Nebenwirkung wurde aber übersehen. Ein schwerwiegender Kühlmittelverlust war von dem „harmlosen“ Kühlmittelverlust nicht mehr klar zu unterscheiden.

Und ein solcher schwerwiegender Kühlmittelverlust ist dann tatsächlich eingetreten. Er wurde nicht rechtzeitig erkannt (Leveson, 1995).

Der Fehlschluss (sowohl der Bedienmannschaft als auch des Wartungsdienstes) lässt sich auf folgendes Schema reduzieren: Mit den Hypothesen „geringer Kühlmittelverlust“ (H_1) und „schwerwiegender Kühlmittelverlust“ (H_2) und der Beobachtung „erhöhte Temperatur im Ablauf“ (E) ergeben sich die Prämissen: $H_1 \leq E$ (lies: H_1 impliziert E), $H_2 \leq E$ und E. Da der Bedienungsmannschaft der geringe Kühlmittelverlust wohl vertraut ist, wird der Schluss auf H_1 gezogen. Das ist ein Zusammenwirken vom eindimensionalem Kausaldenken und verfehlter Induktion.

14.5 Der Lernzyklus

Unsere Vorfahren aus vorgeschichtlicher Zeit konnten sich beim Sammeln und Jagen noch gut gerüstet fühlen. Heute leben wir in einer zum Großteil von uns selbst geschaffenen, künstlichen Welt. Und auf die hin sind wir nicht selektiert worden. Unser biologisches Erbe hat uns nicht von vornherein mit den Fähigkeiten ausgestattet, mit der Technik umzugehen. Wir bauen uns unsere Fallen selbst.

Wir müssen zum Großteil unsere angeborenen Fähigkeiten um Erlerntes und Eingeübtes erweitern, um in der technischen Welt bestehen und die Quellen möglicher Irrtümer erkennen zu können.

Wir haben die Fähigkeit, zu lernen. Die Lehrmeister sind uns glücklicherweise angeboren. Das Problem besteht darin, dass wir erst einmal herausfinden müssen, was zu lernen lohnt.

Zur Demonstration des Lernprozesses wähle ich ein Beispiel aus der Programmierung. Es ist auch von Nichtinformatikern leicht nachvollziehbar.

Wenn dem Anfänger eine Klassifikationsaufgabe mit Fallunterscheidungen vorgelegt wird, neigt er dazu, die Aufgabe als eine Folge von einfachen und voneinander unabhängigen Einzelentscheidungen zu sehen. Die Grundstruktur des sich so ergebenden Programms sieht etwa so aus:

```
IF A THEN ... END;
IF B THEN ... END;
IF C THEN ... END;
...
```

Das Programm durchläuft also nacheinander irgendwelche Entscheidungsknoten, und in jedem Knoten wird zwischen zwei Alternativen gewählt, ohne auf die bisherigen Entscheidungen Bezug zu nehmen. Eine der Alternativen besteht meist gar im Nichtstun (leere Anweisung) wie im obigen Muster, wo die Else-Zweige der Auswahlanweisungen fehlen (Bild 14.1).

Bild 14.1 Entscheidungssequenz

Beispiel Dreiecksklassifizierung: Seien a, b und c natürliche Zahlen, die die Seitenlängen eines Dreiecks repräsentieren mögen. Das Programm soll feststellen, zu welcher Klasse das Dreieck gehört, nämlich ob es sich um ein ungleichseitiges, ein gleichschenkliges oder ein gleichseitiges handelt.

Eine Lösung nach dem Muster der Entscheidungssequenz könnte in der Programmiersprache Oberon (Reiser, Wirth, 1992) folgendermaßen aussehen:

```
IF (a#b) & (a#c) & (b#c) THEN klasse:= ungleichseitig END;
IF (a=b) & (b=c) THEN klasse:= gleichseitig END;
IF (a=b) OR (a=c) OR (b=c) THEN klasse:= gleichschenklig END;
```

Diese Vorgehensweise ist einfach und wird zur Gewohnheit. Aber: die Struktur ist nur vordergründig einfach. Programmiertechnisch ist sie ineffizient, weil unnötige Abfragen ausgeführt werden, und sie ist schwer beherrschbar und fehleranfällig.

Hin und wieder werden Fallunterscheidungen nicht konsequent durchdacht. Es kommt zu einander überlappenden Bedingungen, und zuweilen werden Fälle übersehen. Im eben dargestellten Programm steckt nun tatsächlich ein Fehler - haben Sie ihn bemerkt? Bei Eingabe gleicher Werte für a, b und c ist sowohl die Bedingung der zweiten als auch die der dritten If-Anweisung erfüllt. Beide Then-Zweige werden durchlaufen. Die korrekte erste Zuweisung klasse:= gleichseitig wird durch die dritte Zuweisung klasse:= gleichschenklig überschrieben. Das Programm liefert die zu schwache Aussage, dass das Dreieck gleichschenklig ist. Die stärkere Aussage, dass das Dreieck gleichseitig ist, geht verloren. Vertauscht man im Programm die zweite mit der dritten Zeile, kommt das Richtige heraus.

Wenn dem Programmierer die Fehleranfälligkeit seiner Programmiergewohnheit bewusst wird und wenn er einmal ein gut programmiertes Beispiel kennen gelernt hat, wird er den Entscheidungsbaum als eine bessere Alternative zur Entscheidungssequenz erkennen. Bei der Dreiecksklassifizierung würde er vielleicht auf die folgende Variante kommen:

```
IF (a#b) & (a#c) & (b#c) THEN klasse:= ungleichseitig
ELSIF (a=b) & (b=c) THEN klasse:= gleichseitig
ELSE klasse:= gleichschenklig END;
```

Bild 14.2 zeigt die Grundstruktur des zugehörigen Entscheidungsbaums. Das Programm ist einfacher und effizienter als das vorherige; und es ist korrekt.

Ein solches Programm lässt sich kaum nach der Probiermethode herstellen - es erfordert ein diskursives Vorgehen: Bei jeder Entscheidung muss man sich klar machen, welche logischen Bedingungen im Else-Zweig gelten. Die Chancen für Programmierfehler schwinden. Für die notwendigen Umformungen genügt Vertrautheit mit der Aussagenlogik etwa so weit, wie sie in Kapitel 7 dargestellt ist.

Bild 14.2
Entscheidungsbaum

Wenn der Programmierer erneut auf Klassifizierungsaufgaben stößt, wird er dieses Schema wieder anwenden. Mit der Zeit geht es ihm „in Fleisch und Blut“ über und er wendet das Schema genau so automatisch an, wie ehemals die Entscheidungssequenz. Der Programmierer hat einen *Lernzyklus* (Bild 14.3) durchlaufen.

Schauen wir uns die Phasen des Lernzyklus an: Die Unfähigkeit, gewisse Aufgaben angemessen zu lösen, ist uns zunächst nicht bewusst. Wir wenden automatisch unsere Methoden, Regeln oder Schemata an, die sich in anderen Zusammenhängen bewährt haben, hier jedoch nicht passen (Zustand 1). Irgendwann wird uns diese Unfähigkeit hoffentlich bewusst (Zustand 2). Wir machen uns schlau und gehen zu besseren Lösungswegen über, die wir zunächst ganz bewusst verfolgen müssen (Zustand 3). Mit der Zeit wird uns das Neue zur Selbstverständlichkeit. Wir brauchen nicht mehr groß darüber nachzudenken. Eine neue Stufe des Expertentums ist erreicht (Zustand 4): Das vormals Neue ist zur Routine geworden.

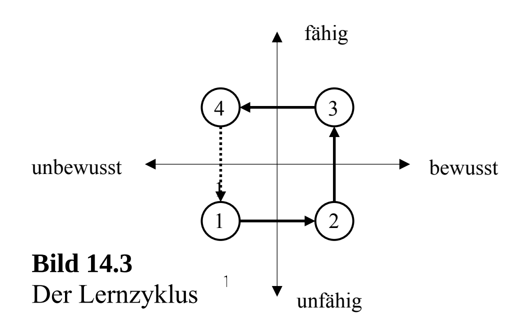


Bild 14.3
Der Lernzyklus

Da sich die Anforderungen an unsere Lösungskompetenz nicht zuletzt wegen der sich wandelnden Technik ändern, kommt es zu neuen Problemen. Unsere Rezepte hören auf, zu passen. Früher oder später landen wir wieder im Zustand der *unbewussten Inkompetenz* (Zustand 1). Der Kreis ist geschlossen. Die vier Stufen des Lernens habe ich von William F. Hayes übernommen (Humphrey, 1995, S. 477 f.).

Der Lernzyklus ähnelt dem Problemlösungsprozess im Rahmen des Generischen Fehlermodellierungssystems. Aber dort geht es um die kurzfristigen Übergänge bei im Grunde unveränderter Problemlösungskompetenz, also darum, Probleme überhaupt als solche wahrzunehmen (Übergang von 1 nach 2) und auf eine höhere Ausführungsebene zu heben (Übergang von Zustand 2 nach 3). Der Lernzyklus soll darüber hinaus verdeutlichen, wie sich die Problemlösungskompetenz verändert (Übergang von Zustand 3 nach 4): Die automatische Ebene wird durch den Lernvorgang angereichert und erweitert - entsprechend verschiebt sich die Grenze zwischen den Ausführungsebenen. Tabelle 14.1 zeigt zusammenfassend die Ebenen der Aufgaben- und Problembewältigung.

Im Zusammenhang mit dem Lernzyklus tauchen drei neue Fragen auf:

1. Wie erfahre ich von meiner Inkompetenz, also von der eigenen Fähigkeitslücke?
2. Wie organisiere ich meinen persönlichen Lernprozess am wirkungsvollsten?
3. Wie sieht eine betriebliche Organisation aus, so dass alle bestmöglich aus den Fehlern lernen können?

Einige Antworten bieten die folgenden Abschnitte.

Tabelle 14.1 Ebenen der Aufgaben- und Problembewältigung

Gegenstand	***Ebenen***		
Klassifizierung: Aufgabe oder Problem?	*Aufgabe:* Es gibt keine Widerstände gegen eine sofortige Lösung.		*Problem:* Es sind Hindernisse zu überwinden.
Ausführungsebene	fähigkeitsbasiert	regelbasiert	wissensbasiert
Art der Aufgaben- und Problembewältigung	automatisch, algorithmisch	auswählend	produktiv, kreativ
Beteiligung des Bewusstseins	vorherrschend unbewusst	mit geringer Aufmerksamkeit	bewusst
Fehlertyp	Schnitzer	Irrtum	

14.6 Die negative Methode

Wie können wir der *unbewussten Inkompetenz* entkommen? Wie schaffen wir den Übergang von Zustand 1 zu Zustand 2 des Lernzyklus und heben die Inkompetenz ins Bewusstsein?

Fehler oder mangelnde Kompetenz können auf dreierlei Wegen bewusst werden (Reason, S. 189 ff.). Nämlich durch

1. Hinweise aus der Umgebung,
2. andere Personen und
3. Selbstüberwachung.

Ein Hinweis aus der Umgebung liegt vor, wenn man den Programmwahlschalter verkehrt herum dreht und daraufhin die Waschmaschine einfach nicht anläuft.

Die Kontrolle durch andere Personen hilft, Fehler zu vermeiden. Diese Kontrolle kann durch die Organisation im Unternehmen erzwungen werden. Eine solche Kontrolle durch andere ist sehr wirkungsvoll und wird beispielsweise für das Testen von Programmen ganz allgemein empfohlen.

Die Selbstüberwachung steht einem immer zur Verfügung. Aber manchmal hilft sie wenig - zum Beispiel dann, wenn bereits das Ziel falsch definiert ist.

Auch kommt es darauf an, die Selbstüberwachung zur rechten Zeit zu aktivieren. Ich erinnere mich an ein „Musterprogramm“ zur Berechnung von Zweierpotenzen. Es lieferte die Werte 2, 4, 7, 15, 32, 63, 127, 255, 511, 1024, 4095, ... Der Programmierer war sich seiner Programmierkunst so sicher, dass er sich den Blick auf das Ergebnis gespart hat. Bei Aktivierung der Selbstüberwachung hätte er den Programmierfehler wohl erkannt. Aber dazu hätte er sich seiner Inkompetenz erst bewusst sein müssen.

Wir sind in einer unschönen Lage: Durch Selbstüberwachung können wir Inkompetenz wahrnehmen, aber die Selbstüberwachung setzt erst ein, wenn einem die eigene Inkompetenz gewahr geworden ist. Es sieht nach einem logischen Zirkel aus: Entweder man hat beides, oder man hat beides nicht.

Dieser logische Zirkel muss aufgebrochen werden. Selbstüberwachung darf nicht erst bei Wahrnehmung der Inkompetenz aktiviert werden!

Die Fähigkeit zur rechtzeitigen Aktivierung und zur erfolgreichen Anwendung der Selbstüberwachung lässt sich lernen und trainieren. Diese Fähigkeit hängt wesentlich davon ab, ob wir Denkfallen erkennen können. Bereits die unterlassene Aktivierung beruht ja auf einer Denkfalle; sie geht auf Selbstzufriedenheit bei unbewusster Gefahr zurück (Complacency-Fehler).

Wenn wir eine Denkfalle erkannt haben, können wir uns dagegen rüsten. Darin sind nämlich die Denkfallen den optischen Täuschungen sehr ähnlich. Ist sie erst einmal bekannt, gibt es auch den Weg, der um sie herumführt - „Forewarned is forearmed“. Bei den optischen Täuschungen reicht es meist aus, ein Lineal anzulegen.

Gegen Denkfallen gibt es kein Einheitsrezept. Wir brauchen eine Liste der Warnzeichen und eine ganze Sammlung von Methoden und Regeln, die uns fit machen. Zu diesen Methoden gehören unter anderen die Methoden der Mathematik und der Logik; aber auch Regelkataloge und Checklisten sind geeignete Gegenmittel. Man muss sie nur im richtigen Moment bereit haben und anwenden.

Die Methoden lassen sich hinreichend konkret nur im Zusammenhang mit bestimmten Technologien, Aufgabengebieten und Tätigkeitsbereichen entwickeln. Aber es gibt eine Reihe von allgemeinen Verhaltensregeln, die für alle Tätigkeitsfelder von Nutzen sind. Ich fasse sie unter der Bezeichnung *negative Methode* zusammen. Sie soll

- das eindimensionale Kausaldenken,
- die Selbstzufriedenheit bei unbewusster Gefahr,
- die Induktionsfehler und
- die unbewusste Inkompetenz

aufdecken und überwinden helfen. Die negative Methode umfasst die folgenden allgemeinen Verhaltensregeln:

1. Warnzeichen für die Unangemessenheit der eigenen Gewohnheiten, Rezepte und Methoden suchen und ernst nehmen.

2. Nicht nach Bestätigung von Vorurteilen und Hypothesen suchen, sondern
3. vor allem nach Gegenbeispielen und Widerlegungen Ausschau halten.
4. Die Ursachen von Fehlern gründlich analysieren.
5. Die Methoden und Techniken der Problemlösung - also die eigene Kompetenz - weiterentwickeln, so dass diese Fehler künftig vermieden werden.

Die Fähigkeit, Warnzeichen und Denkfallen zu entdecken, lässt sich entwickeln und trainieren. Und das geht über

- das Studium von Fehler- und Unfallberichten;
- die Sammlung der dabei im Nachhinein identifizierten Warnzeichen;
- eine Ursachenanalyse mit einer Benennung möglicher Denkfallen;
- das Studium von Vorbildern und die Analyse ihrer Arbeiten.

Den Begriff „negative Methode“ habe ich von Karl R. Popper übernommen, der ihn eher nebenbei einmal verwendet hat, um seinen erkenntnistheoretischen Ansatz der „Betonung der *negativen Argumente* wie Gegenbeispiele, Widerlegungen, Widerlegungsversuche - kurz: Kritik“ zu charakterisieren (Popper, 1973, S. 32). Die negative Methode geht über das reine Lernen aus Fehlern hinaus und erweitert es um die *Suche nach Fehlern*.

14.7 Der Regelkreis des selbstkontrollierten Programmierens

Das Lernen aus den Fehlern, den Lernzyklus, können wir perfektionieren und durch externes Wissen in Form von schriftlich fixierten Regeln ergänzen, wie beispielsweise beim *Regelkreis des selbstkontrollierten Programmierens* (Bild 14.4). Zentrale Elemente dieses Regelkreises sind

- ein Katalog von Programmierregeln und
- die Technik der Fehleranalyse (Analyse der Fehlerursachen).

Die Programmierregeln notiert man sich am besten in einer eigenen Datei. Diese schreibt man im Laufe des Lernprozesses fort (Grams, 1990).

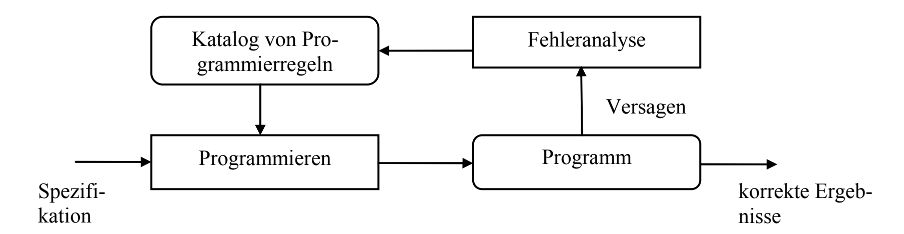


Bild 14.4 Der Regelkreis des selbstkontrollierten Programmierens

Den Katalog von Programmierregeln baut man um einen Kern herum auf, der die allgemein akzeptierten Regeln und Prinzipien für eine gute Strukturierung von Programmen enthält. Die Stichworte: Lesbarkeit, schrittweise Verfeinerung, einfache hierarchische Strukturen, überschaubare Programmmodule mit sparsamen Schnittstellen- und Variablendeklarationen, Verbergen der Funktionsdetails (Information Hiding). Im folgenden Abschnitt werden diese Dinge im historischen Zusammenhang dargestellt.

Dazu kommen diejenigen Regeln, die sich aus eingehenden Fehleranalysen ergeben haben. Für mich haben auf diese Weise die folgenden Regeln Bedeutung gewonnen:

- Benutze Entscheidungsbäume beim Klassifizieren und beim Aufstellen komplexer logischer Bedingungen.
- Wähle für Variablen und Prozeduren Namen, die deren Bedeutung und Funktion möglichst exakt bezeichnen.

Die erste der Regeln verhindert die Blickverengung und einen Reinfall auf das eindimensionale Ursache-Wirkungsdenken. Im Abschnitt 14.5 wurde gezeigt, wie eine solche Regel hat entstehen können.

Die zweite Regel unterstützt den Wahrnehmungs- und Denkapparat bei seiner Sinnsuche. Die zweite Regel entstand, als ein Programmierfehler darauf zurückzuführen war, dass eine Funktion - ich nannte sie „equivalent“ - doch nicht genau das tat, was ihr Name versprach - nämlich die Äquivalenz zweier Objekte festzustellen.

Man kann die Arbeit am Regelkatalog mit einem Kondensationsprozess vergleichen: Es geht um die *Verdichtung von Information*. Der Katalog sollte nicht zu viele Regeln enthalten. Und alle müssen aussagestark sein. Die Hauptrolle bei der Erstellung und der Perfektionierung des Regelkatalogs spielt die Fehleranalyse. Dabei wird auch der Frage nachgegangen, inwieweit die Wahrnehmungs- und Denkgesetze am Zustandekommen der Programmierfehler beteiligt sind (Grams 1990).

14.8 Strukturierungstechniken - 50 Jahre Programmierung

Im Laufe der Entwicklung von Programmiersprachen sind wichtige Regeln und Prinzipien der Programmentwicklung entstanden. Das soll im folgenden historischen Abriß deutlich werden.

Die Anfänge - im Gefühl der Allmacht in die Krise

In den 50-er Jahren wird in Assembler oder einer problemorientierten Programmiersprache wie FORTRAN (1954) und COBOL (1959) programmiert. Die Mittel erlauben viel, verbieten fast nichts und ermutigen zum Schritt vom „Programmieren im Kleinen“ zum „Programmieren im Großen“. Der Glaube, dass man nur genügend Manpower brauche, um jedes Problem lösen zu können, führt geradewegs in die Software-Krise. Es entstehen fehlerbehaftete Programme mit unübersehbarem Wirkungsgefüge („Spaghetti-Codes“).

Strukturierte Programmierung

Die erste Stildiskussion in den 60-er Jahren bringt die *strukturierte Programmierung* (Dahl/ Dijkstra/Hoare, 1972): Die Gestaltungsbeliebigkeit von Programmiersprachen wird durch die Beschränkung auf die *Strukturblocktypen* Sequenz, Auswahl und Schleife eingeschränkt. Es kommt zur Ächtung der freien Sprünge (Goto-freies Programmieren). Der Übergang zu *formatfreien* Sprachen erlaubt die Sichtbarmachung der Programmstruktur durch Einzugstechnik (Indentation).

Als wesentliche Entwurfsmethode setzt sich die *schrittweise Verfeinerung* durch (Top-Down-Entwurf): Ausgehend von der Spezifikation wird die Aufgabe in Teilaufgaben zerlegt, für die wiederum Spezifikationen erstellt werden. Schließlich erhält man Aufgaben, die sich durch elementare Programmbausteine lösen lassen. Ein weiteres wichtiges Mittel zur Programmstrukturierung und zur Vermeidung von Redundanz ist das *Prozedurkonzept*.

Durch strenge *Typisierung* können falsche Wertzuweisungen bereits bei der Übersetzung eines Programms aufgedeckt werden. Als theoretischer Überbau entsteht die Mathematisierung und Axiomatisierung der Programmierung.

Ergebnis der Entwicklung sind die Programmiersprachen ALGOL (1958), PL/I (1965) und Pascal (1970). Zu dieser Klasse von Programmiersprachen gehört auch das heute weit verbreitete C (1974). Durch Aufweichung einiger Regeln der strukturierten Programmierung wird bei dieser Sprache eine größere Flexibilität erreicht. Das geht zu Lasten der Beherrschbarkeit.

Modulare Programmierung

Mit umfangreicher werdenden Programmieraufgaben stößt die Erstellung monolithischer Programme an Grenzen: Aufgabenzerlegung und -aufteilung auf mehrere Programmierer und Programmierteams sowie die nun notwendige Verantwortungsabgrenzung führen in den 70-er Jahren zur *modularen Programmierung:* Die Programmiersprachen unterstützen die Erstellung von separat übersetzbaren Programm-Modulen mit öffentlichen Schnittstellen, deren Innenleben vor dem Anwender des Moduls weitgehend verborgen werden kann (Information-Hiding). Dieses Konzept wird in den Programmiersprachen MODULA-2 (1977) und Ada (1979) umgesetzt.

Strukturierte und modulare Programmierung sind geprägt durch das reine Top-Down-Vorgehen. Es ist gekennzeichnet durch das *Denken vom Resultat her:* Alle Aktivitäten sind darauf ausgerichtet, die Spezifikation des Systems zu erfüllen. Dieser Ansatz führt zu relativ steifen Systemen, und nimmt nicht Rücksicht auf die Wiederverwendbarkeit der Programmteile.

Objektorientierte Programmierung

Die Anstrengungen zur Überwindung dieser Beschränkungen führen zu Beginn der 80-er Jahre zum neuen Programmier-Paradigma der *objektorientierten Programmierung.* Daten und Methoden (Prozeduren und Funktionen) werden zu Objekttypen (Klassen) zusammengefasst. Ein Vererbungskonzept sowie die Einführung dynamischer Datentypen ermöglichen weitgehende *Datenabstraktion*. Der Trick, die Wiederverwendbarkeit zu erreichen, lässt sich mit dem „Trick" der Algebra vergleichen: abstrakte Klassen in der objektorientierten Programmierung sind in ähnlicher Weise konkretisierbar und „wiederverwendbar" wie die abstrakten Strukturen Ring, Körper und Vektorraum der Algebra.

Neben den Top-Down-Entwurf tritt in der objektorientierten Programmierung der Bottom-Up-Entwurf. Der Trend zur Wiederverwendung von Komponenten ändert den Charakter der Programmiertätigkeit: die Entwicklung geht vom Schreiben hin zum Lesen. *Vertragsbasiertes Programmieren* (Programming by Contract) soll das korrekte Zusammenspiel der Komponenten sicherstellen (Meyer, 1988): Für die Klassen und deren Methoden wird die Spezifikation unter Angabe von Vor- und Nachbedingung explizit gemacht. Die Vorbedingung sagt, unter welchen Bedingungen die Komponente (ein Modul, eine Klasse oder eine Funktion) ihre Leistung erbringt. Die Vorbedingung ist also derjenige Vertragsbestandteil, der den Anwender eines Moduls bindet. Die Nachbedingung präzisiert, welches Ergebnis die Komponente liefert. Eine herausragende Rolle bei der Software-Konstruktion spielen die Klasseninvarianten; sie sind Vor- und Nachbedingung einer jeden Methode einer Klasse.

Die meisten der Elemente der objektorientierten Programmierung sind zwar bereits in der frühen Programmiersprache SIMULA (1967) realisiert. Richtig zur Geltung kommen sie aber erst mit den Sprachen Smalltalk 80 (1980), Eiffel (1986), C++ (1986), Oberon (1988) und Java (1996).

In der Praxis hat C++ wohl die größte Bedeutung erlangt. Durch die Kompatibilität zu C, die Erweiterung um die Objektorientierung und durch Addition einiger eher unnötigen Sprachelemente hat die Sprache C++ einen kaum mehr beherrschbaren Umfang angenommen.

Grafische Repräsentation und direkte Manipulation

Zunehmende Bedeutung gewinnen Programmierumgebungen und Programmierhilfsmittel, die mit grafischen Repräsentationen arbeiten und die auf einer höheren Ebene die Beherrschbarkeit der Sprachen wieder herstellen sollen. Ein solches programmiersprachenunabhängig einsetzbares Hilfsmittel ist die *vereinheitlichte Modellierungssprache* (Unified Modeling Language) UML (Balzert, 1999).

In den 90-er Jahren wird besondere Aufmerksamkeit der Entwicklung von grafischen Bedienoberflächen gewidmet. Trotz Objektorientierung ist die textuelle Erstellung von Oberflächen mühsam. Durch den Übergang zur *visuellen Programmierung* wird diese Arbeit wesentlich erleichtert: Drag-and-Drop-Techniken ermöglichen die direkte Manipulation der Elemente einer grafischen Benutzeroberfläche (GUI, Graphical User Interface). Während der Entwicklung sieht man die Oberfläche entstehen (What you see is what you get, WYSIWYG).

Die visuelle Programmierung ist eher ein Merkmal der Entwicklungsumgebung als eines der Programmiersprache selbst. Eine der ersten weit entwickelten Programmierumgebungen dieser Art ist Delphi auf der Basis von Object Pascal (1995). Heute gibt es solche Programmierumgebungen für alle gängigen objektorientierten Sprachen.

14.9 Halbformale und formale Konstruktionsmethoden

Beweisgeleitetes Programmieren

Beim *beweisgeleiteten Programmieren* geht man von einer möglichst exakten Formulierung von Vor- und Nachbedingung des Systems - beispielsweise eines Algorithmus - aus. Diese mathematische Formulierung wird Schritt für Schritt mittels Äquivalenztransformationen in eine Form gebracht, die sich mittels Programm einfach realisieren lässt.

Beispiel: Gesucht ist eine Goldversion für die Orthogonal-Aufgabe (Abschnitt 1.6). Die Menge der Eingabedaten umfasst sämtliche Paare ganzer Zahlen (α, β). Da α und β beliebige Zahlen sein dürfen, ist auch für die Differenz $\alpha - \beta$ jeder ganzzahlige Wert möglich.

Rechtwinkligkeit kann nur vorliegen, wenn diese Differenz ein Vielfaches von 90 beträgt, wobei Vielfache von 180 auszuschließen sind. Das gewünschte Resultat lässt sich also folgendermaßen fassen: Orthogonal soll genau dann den Wert true haben, wenn mit einer ganzen Zahl k gilt, dass $\alpha\text{-}\beta = 90k$ ist, und dass andererseits keine Zahl j existiert, so dass $\alpha\text{-}\beta = 180j$ gilt.

Der zweite Teil dieser Spezifikation besagt: Die Zahl k aus dem ersten Teil der Spezifikation muss ungerade sein. Das heißt, dass k mit einer ganzen Zahl i die Darstellung $k = 2i + 1$ gestattet. Mit dem so definierten k ergibt sich eine etwas kürzere Formulierung des angestrebten Resultats: Orthogonal soll genau dann den Wert true annehmen, wenn es eine ganze Zahl i gibt, so dass $\alpha\text{-}\beta\text{-}180i = 90$.

Bekanntlich ist n mod m gleich dem Wert $n - m \cdot i$, wobei i so eingerichtet wird, dass das Ergebnis nichtnegativ und kleiner m ist: $0 \leq n - m \cdot i < m$. Damit lässt sich das gewünschte Resultat noch prägnanter formulieren: Orthogonal soll genau dann wahr sein, wenn $(\alpha\text{-}\beta)$ mod 180 = 90. Und in dieser Form lässt sich die Formel direkt in ein Programm umsetzen. In Pascal sieht -

mit Rücksicht auf weit verbreitete Compiler, die den Modulo-Operator nicht korrekt implementieren - die Lösung so aus:

```
FUNCTION Orthogonal(Alpha, Beta: INTEGER): BOOLEAN;
BEGIN Orthogonal:= abs(Alpha-Beta) MOD 180 = 90 END;
```

Die entsprechende C-Funktion (Kernighan/Ritchie, 1988) ist

```
int Orthogonal(int Alpha, int Beta){return abs(Alpha-Beta)%180==90;}
```

Dieses „Programm" ist nicht nur kürzer als die auf intuitive Weise gewonnenen, es verdient auch größeres Vertrauen in seine Korrektheit. Die Vertrauenswürdigkeit liegt an der Art der Herleitung: Schritt für Schritt ging es von der Problemstellung zum Programm. Jeder Teilschritt, jede Umformung war klein und überschaubar, wie bei einem mathematischen Beweis. Das ist die diskursive Methode.

Zu den speziellen Techniken der diskursiven Programmierung gehört die Verwendung von Invarianten. Sie bieten Hilfe bei der Konstruktion von Algorithmen und Datenstrukturen.

Invarianten zur Konstruktion von Algorithmen

Eine dominierende Grundstruktur vieler Algorithmen ist die Schleife (Iteration). Als ausgesprochen hilfreich bei der Konstruktion solcher schleifenbasierter Algorithmen haben sich die *Schleifeninvarianten* erwiesen. Eine Schleifeninvariante macht Aussagen über die Relationen, die zwischen den Variablen des Algorithmus bestehen. Die Schleifeninvariante ist sozusagen das Gesetz, dem der Algorithmus unterworfen wird: Wenn die Invariante vor Eintritt in den Schleifenkörper gilt, dann soll sie auch unmittelbar nach Ausführung der Anweisungen des Schleifenkörpers gelten.

Die Kunst ist nun, eine Aussage zu finden, die das Wesentliche des Algorithmus erfasst, die also möglichst viel aussagt. Vielsagend soll die Invariante sein, weil sie ja bei der Konstruktion des Programms den Weg weisen soll. Ein einfaches Beispiel soll das Arbeiten mit Invarianten verdeutlichen.

Intervallschachtelung: Die Aufgabe lautet, eine Nullstelle x der stetigen reellen Funktion $f(x)$ zu finden, die für alle x von 0 bis 1 definiert ist. Vorausgesetzt wird, dass $f(0) < 0 \leq f(1)$ gilt. Aus dem Zwischenwertsatz der Analysis folgt, dass eine Nullstelle im Intervall von 0 bis 1 liegen muss. Wir versuchen, die Nullstelle immer stärker einzugrenzen, indem wir die Unter- und die Obergrenze des Intervalls, in dem eine Nullstelle liegen muss, immer näher aneinanderrücken.

Die (variable) Untergrenze bezeichnen wir im Programm mit a und die Obergrenze mit b, die Funktion selbst mit f. Als Invariante nehmen wir die Aussage f(a)<0≤f(b). Wenn diese Aussage gilt, muss die Funktion f eine Nullstelle im Intervall von a bis b haben. Die Invariante können wir wahr machen, indem wir den Variablen a und b die Werte 0 und 1 zuweisen. Wenn wir nun das Intervall immer kleiner machen und dafür sorgen, dass auch nach jeder Verringerung die Invariante gilt, haben wir ein wirkungsvolles Instrument zur korrekten Realisierung der Nullstellensuche mittels Intervallschachtelung. In der Programmiersprache C könnte die Schleife etwa so aussehen:

```
a=0; b=1;
while (eps<b-a) {p=(a+b)/2; if (f(p)<0) a=p; else b=p;}
```

In Tabelle 14.2 ist der Ablauf detailliert dargestellt. An verschiedenen Stellen im Programm werden Aussagen über die Programmvariablen und deren Werte gemacht. Solche Aussagen

heißen *Zusicherungen* (Assertion). Die Invariante tritt an verschiedenen Stellen des Programms als Zusicherung auf. Auch das Resultat wird in Form einer Zusicherung beschrieben.

Beim beweisgeleiteten Programmieren beginnt man mit den Zusicherungen. Dann werden die Programmstücke eingefügt und die Beweise geführt, dass die Zusicherungen tatsächlich stimmen. Programmbeweis und Programm werden Hand in Hand entwickelt, wobei der Beweis den Weg weist.

Der Nachweis, dass die Zusicherungen gültig sind, ist für das Beispiel noch zu führen. Zur einfachen Darstellung schreiben wir die durch das Programm tatsächlich veränderlichen Variablen als Folge: (a, b, p). Eine Wertebelegung (auch: Zustand) lässt sich dann durch eine entsprechende Folge von Werten darstellen. Beispielsweise ist der Zustand unmittelbar nach der Initialisierung gegeben durch (0, 1, ?), mit einem undefinierten Wert für die Variable p.

Im Zustand (0, 1, ?) ist die Invariante wahr. Voraussetzungsgemäß gilt nämlich $f(0)<0\leq f(1)$. Da wir Gleiches durch Gleiches ersetzen dürfen, darf die 0 durch a und die 1 durch b ersetzt werden. Damit haben wir die Gültigkeit der Invarianten gezeigt: $f(a)<0\leq f(b)$.

Unmittelbar vor Eintritt in den Schleifenkörper gilt zusätzlich die Schleifenbedingung, denn ansonsten würde der Schleifenkörper gar nicht erreicht. Invariante und Schleifenbedingung zusammen bilden die Vorbedingung des Schleifenkörpers.

Tabelle 14.2 Programmablauf und Zusicherungen eines als Schleife realisierten Algorithmus. Die grau unterlegten Zellen enthalten Zusicherungen. Sie gehören nicht zum ausführbaren Programm.

Bezeichnung	***Beispiel***	***Beschreibung***
Initialisierung	`a=0; b=1;`	Die Invariante wird wahr gemacht.
Invariante	`f(a)<0; 0≤f(b)`	Die Invariante gilt vor Schleifeneintritt.
Schleifenbedingung	`eps<b-a`	Falls die Schleifenbedingung wahr ist, wird fortgefahren, ansonsten wird die Ausführung der Schleife durch Sprung zum Ende abgeschlossen.
Vorbedingung für Schleifenkörper	`f(a)<0; 0≤f(b);` `eps<b-a`	Invariante und Schleifenbedingung sind gültig.
Schleifenkörper	`p=(a+b)/2;` **`if`** `(f(p)<0)a=p;` **`else`** `b=p;`	Die Anweisungen werden ausgeführt.
Nachbedingung für Schleifenkörper	`f(a)<0; 0≤f(b)`	Invariante ist gültig. Damit ist - nach einem Rücksprung - die Vorbedingung wieder erfüllt.
Rücksprung		Sprung zur Schleifenbedingung.
Ende: Bedingung für das Resultat	`f(a)<0; 0≤f(b);` `b-a≤eps`	Invariante und negierte Schleifenbedingung sind gültig.

Zu zeigen ist, dass nach Durchlaufen des Schleifenkörpers die Invariante wieder gilt (Nachbedingung). Es genügt nun allerdings nicht, den Nachweis nur für eine bestimmte Wertebelegung zu führen. Wir müssen ganz allgemein beweisen, dass aus der Gültigkeit der Vorbedingung nach Abarbeitung des Schleifenkörpers sich die Nachbedingung ergibt. Das ist der Kern aller Korrektheitsbeweise für Programme.

Den Korrektheitsbeweis für den Schleifenkörper führen wir gemäß der Methode der *symbolischen Ausführung* in folgenden Teilschritten durch.

1. Anfangsbelegung und Übersetzung: Anstelle der konkreten Zahlenwerte für die Variablen belegen wir die Programmvariablen mit Variablen im mathematischen Sinn - also mit Variablen, deren Werte zwar beliebig gewählt aber fest sind. Die Vorbedingung des Programmabschnitts wird in die Sprache der Mathematik übersetzt, indem die Programmvariablen durch ihre momentanen Werte, also die jeweiligen mathematischen Variablen ersetzt werden.

2. Symbolische Programmausführung: Dann führen wir die Anweisungen des zu untersuchenden Programmabschnitts Schritt für Schritt mit diesen mathematischen Variablen - also symbolisch - aus. Als Endzustand ergibt sich eine bestimmte Wertebelegung der Programmvariablen.

3. Mathematische Umformungen: Nun stehen auf den Programmvariablen Ausdrücke, in denen nur noch diese mathematischen Variablen vorkommen. Der Gültigkeitsnachweis für irgendwelche Aussagen findet ausschließlich im Bereich der Mathematik statt.

4. Rückübersetzung: Schließlich werden die mathematischen Variablen wieder aus den mathematischen Beziehungen eliminiert, indem die Programmvariablen anstelle ihrer nun gültigen Werte eingesetzt werden. Dadurch ergeben sich die am Ende gültigen Zusicherungen. Der Programmbeweis ist gelungen, wenn diese Zusicherungen die Nachbedingung implizieren.

Der Korrektheitsbeweis für den Schleifenkörper unseres Beispiels geht nach diesem Rezept folgendermaßen vor sich:

1. Schritt (Anfangsbelegung und Übersetzung): Anfangs ist die Wertebelegung für (a, b, p) gegeben durch (A, B, P). Dabei folgen wir der Konvention, dass wir die Programmvariablen mit Kleinbuchstaben und die mathematischen Variablen mit Großbuchstaben bezeichnen. Da anfangs die Invariante - und zusätzlich die hier nicht weiter interessante Schleifenbedingung - als gültig vorausgesetzt wird, gilt $f(a)<0\leq f(b)$. Wir ersetzen die Programmvariablen durch ihre derzeit gültigen Werte. Daraus folgt $f(A)<0\leq f(B)$. Damit haben wir die Vorbedingung in die Sprache der Mathematik übersetzt.

2. Schritt (Symbolische Programmausführung): Wir verfolgen nun die Wertebelegungen bei Abarbeitung des Schleifenkörpers: Nach der Zuweisung „p=(a+b)/2;“ ist die Wertebelegung gleich (A, B, (A+B)/2). Falls $f((A+B)/2)<0$ ist, geht es mit der Anweisung „a=p;“ weiter, was zum Endzustand ((A+B)/2, B, (A+B)/2) führt. Vorerst betrachten wir nur diesen einen Fall und merken uns, dass der andere Fall der Verzweigung noch offen steht.

3. Schritt (Mathematische Umformung): Wir bringen die mathematischen Relationen in eine Form, so dass die spätere Rückübersetzung möglichst einfach wird. Hier genügt es, die gültigen mathematischen Beziehungen geeignet hinzuschreiben: $f((A+B)/2)< 0 \leq f(B)$.

4. Schritt (Rückübersetzung): Wir eliminieren die mathematischen Variablen, indem wir wieder die Programmvariablen einsetzen. Da a nach Durchlaufen des Programmabschnitts den

Wert (A+B)/2, und b den Wert B hat, können wir im Ausdruck diese Werte durch die Variablen ersetzen: $f(a) < 0 \leq f(b)$. Das heißt: Nach Durchlaufen des Schleifenkörpers ist die Nachbedingung gültig.

Ganz analog führt man den Beweis für den Fall, dass $f((A+B)/2)<0$ nicht gilt. Offenbar ist dann $0 \leq f((A+B)/2)$. Da anstelle der Zuweisung „a=p;" nun die Zuweisung „b=p;" abgearbeitet wird, ist die Wertebelegung unmittelbar hinter dem Schleifenkörper gleich (A, (A+B)/2, (A+B)/2). Die Rückübersetzung führt auf dasselbe Ergebnis wie im ersten Fall. Also ist in jedem Fall die Nachbedingung im Anschluss an die Abarbeitung des Schleifenkörpers gültig. Der Programmbeweis ist geführt.

Für jeden möglichen Pfad durch einen zu beweisenden Programmabschnitt ist der Korrektheitsbeweis zu führen. Alle möglichen Pfade durch einen Programmabschnitt lassen sich mit dem Kontrollflussgraphen ermitteln. Auch der klassische Programmablaufplan (Flowchart) erfüllt den Zweck (Balzert, 1999).

Der *Kontrollflussgraph* ist ein Diagramm der Ablaufstruktur, in dem die Anweisungen oder Anweisungssequenzen durch Knoten repräsentiert werden. Die Pfeile dienen zur Darstellung von Übergängen zwischen Programmteilen. In reduzierter Darstellung können Sequenzen von Knoten auch weggelassen werden. Dann sind die Anweisungen dem entsprechenden Pfeil zugeordnet. Bild 14.5 zeigt einen solcherart reduzierten Kontrollflussgraphen für das Beispiel der Intervallschachtelung.

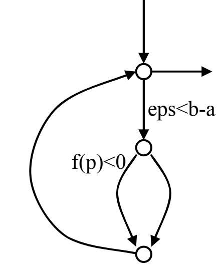


Bild 14.5 Kontrollflussgraph „Intervallschachtelung"

Die Methode der symbolischen Ausführung ist leicht zu verstehen und nach den gängigen Regeln der Mathematik zu praktizieren. Aber sie kann zu erheblichem Rechenaufwand und unübersichtlichen Rechengängen führen - insbesondere dann, wenn viele Fallunterscheidungen und Programmpfade zu beachten sind.

Die *Methode von Hoare* ist ein wesentlich leistungsfähigeres Instrument als die symbolische Programmausführung (Gries, 1981). Leider kommt sie der Intuition nicht gleichermaßen entgegen.

Invarianten beim Aufbau und Löschen von Datenstrukturen

Die Verwaltung komplexer Datenstrukturen in Rechnern bietet eine Reihe von Schwierigkeiten und es besteht die Gefahr, in die eine oder andere Falle zu tappen. Nehmen wir als Beispiel die rechnerinterne Darstellung von Bäumen in der sogenannten *Parent-Repräsentation*. In dieser Darstellungsart sind die Zeiger zwischen den Knoten von den Kindern zum Elter gerichtet. Bild 14.6 zeigt einen Baum mit fünf Knoten.

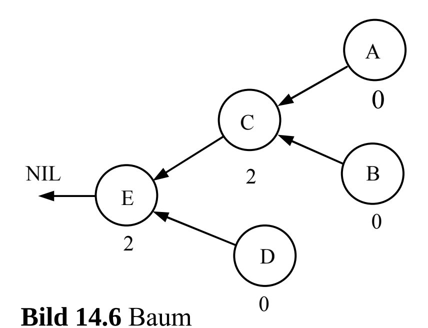


Bild 14.6 Baum in Parent-Repräsentation

Die Parent-Repräsentation hat den Vorteil, dass jedes Objekt nur einen Zeiger benötigt. Jeder *Knoten* wird also durch ein *Objekt* realisiert, das einen Zeiger besitzt, der auf einen Knoten zeigen kann. Wäre man umgekehrt verfahren, hätte jeder Knoten mit einer variablen Anzahl von Zeigern für die Kinder ausges-

tattet werden müssen.

Wir bezeichnen einmal den Zeiger, den jedes Objekt hat und der auf ein anderes Objekt zeigen kann mit next. Im Beispiel ist A.next gleich C, B.next gleich C, C.next gleich E und D.next gleich E. Der Zeigerwert NIL bedeutet, dass der betreffende Zeiger ins Leere zeigt. Das ist bei der Baumwurzel E der Fall: E.next ist gleich NIL.

Ein Problem entsteht, wenn die Struktur, oder ein Teil davon, zu löschen ist. Das Löschen geschieht typischerweise von den Blättern aus. Jedem Objekt möge für die Selbstlöschung die Methode (Prozedur) destroy zur Verfügung stehen: A.destroy entfernt also das Objekt A aus dem Speicher. Üblicherweise entfernt es aber auch gleichzeitig das mit ihm über next verbundene Objekt. Das heißt, dass die Methode destroy vor der Freigabe des Speichers noch das verbundene Objekt über Aufruf des Befehls next.destroy löscht (rekursive Speicherfreigabe). Der Aufruf von A.destroy bewirkt also rekursiv die Löschung der drei Knoten E, C und A in genau dieser Reihenfolge.

Nun zeigen B.next und D.next nicht mehr auf gültige Objekte. Aber die Zeiger haben auch nicht den Wert NIL. Damit ist die gesamte Datenstruktur zerstört: Ein Aufruf von D.destroy beispielsweise würde ein weiteres Mal versuchen, den Knoten E zu löschen - aber der ist ja schon weg. Schwerwiegende Fehlfunktionen des Programms sind die Folge.

Um dem Problem zu begegnen, erhält jeder Knoten ein Attribut namens nref. Dieses Attribut enthält die Anzahl derjenigen next-Referenzen, die auf den Knoten zeigen. Und jetzt wird einfach ausgeschlossen, dass ein Knoten gelöscht wird, dessen nref-Wert größer als eins ist. Im obigen Beispiel gilt A.nref=0, B.nref=0, C.nref=2, D.nref=0 und E.nref=2. Die nref-Werte stehen unter den Knoten.

Jetzt haben wir es mit einer nicht mehr ganz trivialen Programmieraufgabe für die verschiedenen Methoden zum Erzeugen, Einfügen, Entfernen und Löschen von Knoten zu tun. Eine bewährte Möglichkeit ist, dass sich der Programmierer an *Invarianten* orientiert. Das sind die von ihm selbst geschaffenen Gesetze, denen sämtliche Methoden der Klasse der Objekte gehorchen müssen. Im vorliegenden Fall könnte er für die Objekte folgende *Invariante* formulieren:

Für jeden Knoten gilt, dass nref gleich der Anzahl von next-Referenzen ist, die auf ihn verweisen, und dass next genau dann gleich NIL ist, wenn next nicht auf einen Knoten zeigt.

Der Programmierer muss darauf achten, dass alle Knoten-Methoden die Invariante aufrecht erhalten: Gilt die Invariante vor Aufruf einer Methode, dann muss die Invariante auch danach gelten. Die Invariante ist also Vor- und Nachbedingung aller Methoden der Knoten dieser Klasse. Deswegen wird sie auch *Klasseninvariante* genannt (Meyer, 1988). Die Invariante ist Leitschnur bei der Programmierung. Sie hilft, Fehler zu vermeiden. Eine Methode zur bedingten Freigabe von Speicherplatz, welche die Klasseninvariante erhält, sieht in Object Pascal (Kaiser, 1997) so aus:

```
{recursive}procedure Tlink.free;            //Bedingte Speicherfreigabe
begin
  if nref=0 then begin          //Falls Objekt nicht referenziert wird,
    if next<>nil then begin       //und falls ein Nachfolger existiert,
      Dec(next.nref);             //reduziere dessen Referenzzähler und
      next.free                  //veranlasse ihn zur Speicherfreigabe.
    end;
    Destroy                                 //eigene Speicherfreigabe
  end
end;
```

Invarianten können - wie hier geschehen - umgangssprachlich beschrieben werden. Oft ist eine strengere Darstellungsweise angebracht. Das Zustandsdiagramm ist eine davon (Humphrey,

1995). Zu den formalen Beschreibungsmitteln gehören die Aussagen- und die Prädikatenlogik sowie die Mathematik der Mengen und Relationen (Gries, 1981; Linger/Mills/Witt, 1979).

Begriffsbestimmungen (Glossar)

Die Begriffsbestimmungen entsprechen weitgehend denen der VDI/VDE-Richtlinie 3542, Blatt 4. Sie sind, zusammen mit der Übersetzung ins Englische, im Hypertext

`http://www.fh-fulda.de/~grams/Reliability/R&S-Terms.html`

erfasst. Kursivschrift verweist auf weitere Begriffsbestimmungen.

Anforderungen

Beschreibung von Eigenschaften, die das System haben soll. Beabsichtigte *Funktion*$_1$.

Ausfall

Verlust der Fähigkeit eines Systems, bei Einhaltung spezifizierter Bedingungen die geforderte *Funktion*$_1$ zu erfüllen. Bezieht sich nur auf Hardware, also auf Objekte, die grundsätzlich in mehreren Exemplaren nach einem Muster hergestellt werden können und die ihre körperliche Beschaffenheit ändern und somit (zu unterschiedlichen Zeiten) ausfallen können. Das Ereignis "Ausfall" markiert den Zeitpunkt des Übergangs von der *Korrektheit* zu einem *Fehler*$_2$. Siehe auch unter *Versagen*.

Ausfallrate

Betrag der Ableitung des natürlichen Logarithmus der *Überlebenswahrscheinlichkeit* (bezogen auf den *Ausfall*). Bei exponentialverteilter Lebensdauer ist die Ausfallrate gleich dem Kehrwert der mittleren *Lebensdauer*.

Bedienfehler

Ein bedienerseitiger Verstoß gegen die *Spezifikatio*n. Allgemeiner: Eine vermeidbare *Risiko*erhöhung durch den Bediener.

Betriebsbewährtheit

Als betriebsbewährt (auch: erprobt) gilt eine Betrachtungseinheit dann, wenn sie im Wesentlichen unverändert über einen ausreichenden Zeitraum in zahlreichen verschiedenen Anwendungen betrieben wurde und dabei keine oder nur unwesentliche *Fehler*$_3$ festgestellt wurden.

Defekt

Fehler$_2$. Siehe auch *Korrektheit*.

Denkfalle

Das Hintergrundwissen ist einer Aufgabenstellung oder einer zu meisternden Situation nicht angemessen. Hintergrundwissen ist Wissen, das von einer größeren Gruppe - beispielsweise allen Menschen einer Zivilisation - geteilt wird. Eine Denkfalle wird offenbar, wenn ein *Irrtum* in der betrachteten Gruppe weit verbreitet ist.

Deterministisches System

Ein System S, dessen *Funktion*$_1$ deterministisch, also eine Funktion im mathematischen Sinn ist (*Funktion*$_2$). In diesem Fall gibt es zu jedem zulässigen Eingabedatum x genau ein Ausgabedatum y und man schreibt $y = S(x)$

Diversität

Ungleichartige technische Mittel zur Erreichung nützlicher *Redundanz* (z.B. andere physikalische Prinzipien, andere Lösungswege der gleichen Aufgabe).

Entwurfsfehler

Ein auf fehlerhaften Hardware- bzw. Software-Entwurf zurückführbarer *Fehler*$_3$.

Fail-Safe-Verhalten

Sicherheitsgerichtete Fehlerauswirkung. Anwendbar auf technische Anlagen, die einen eindeutigen sicheren Zustand (energielos, ruhend usw.) besitzen. Im Fehlerfall wird die Anlage durch die Steuerungseinrichtung in diesen gesicherten Zustand überführt.

Fehler$_1$

Abweichung zwischen dem berechneten, beobachteten oder gemessenen Wert (einer Ausgangsgröße) und dem wahren, spezifizierten oder theoretisch richtigen Wert aufgrund eines *Fehlers*$_2$ oder einer *Störung*$_1$. Ähnlich *Versagen*.

Fehler$_2$

Nichterfüllung der Spezifikation. Unkorrektheit, *Defekt*. Ursachen: *Ausfall* oder *Fehler*$_3$.

Fehler$_3$

Abweichung der tatsächlichen von der für die Erfüllung der *Spezifikation* erforderlichen konstruktiven und fertigungstechnischen Ausführung des Systems (Verdrahtung, Dimensionierung, Programmierung usw.). Entwurfs-, Programmier- oder Fertigungsfehler. Mögliche Ursache: *Fehler*$_4$.

Fehler$_4$

Menschliche Handlung mit unerwünschtem Ergebnis. *Irrtum* oder *Schnitzer*.

Fehlermaskierung

Verhinderung der Auswirkung eines *Fehlers*$_2$ in einem Subsystem, so dass die ununterbrochene Erfüllung der Funktion$_1$ des Gesamtsystems ermöglicht wird.

Fehlertoleranz

Fähigkeit eines Systems, auch mit einer begrenzten Zahl fehlerhafter Subsysteme seine *Spezifikation* zu erfüllen.

Fehlerwahrscheinlichkeit

Unkorrektheitswahrscheinlichkeit

Funktion$_1$ (eines Systems)

Funktion im weiteren Sinn. Tätigkeit, Wirksamkeit, Verrichtung (Duden).

Ein-/Ausgaberelation S eines Systems: Es ist $(x, y) \in S$ genau dann, wenn y eine mögliche Antwort des Systems auf die (zulässige) Eingabe x ist.

Die Größen x und y können (zeitabhängige) Vektoren sein, aber auch Zeitfunktionen mit beliebigem Wertebereich sind grundsätzlich zugelassen. Die Ein-/Ausgabevariablen sollen alle notwendigen Daten zur Beschreibungen der Interaktion des Systems mit seiner Umgebung enthalten.

Funktion$_2$

Funktion im engeren (mathematischen) Sinn. Rechtseindeutige Relation. Eine Relation F ist rechtseindeutig, wenn aus $(x, y_1) \in F$ und $(x, y_2) \in F$ folgt, dass $y_1 = y_2$. In diesem Fall schreibt man üblicherweise $y = F(x)$.

Funktionale Spezifikation

Spezifikation der normalen Funktion$_1$ im Unterschied zur *Sicherheitsspezifikation*.

Funktionsfähigkeit

Eignung einer Einheit, eine geforderte Funktion unter vorgegebenen Anwendungsbedingungen zu erfüllen. *Korrektheit*.

Gefahr$_1$

Drohender Schaden (ohne Rücksicht auf Höhe und Häufigkeit, Bedeutung wie beispielsweise in §§ 446, 447 und 694 BGB).

Gefahr$_2$

Sachlage, bei der das Risiko größer als das Grenzrisiko ist, wobei unter Grenzrisiko das größte noch vertretbare *Risiko* verstanden wird. Komplementär zu *Sicherheit*.

Gefährdungshaftung

Siehe Produkthaftung.

HRA

Human Reliability Analysis. *Zuverlässigkeit*sanalyse unter Einbeziehung der "Komponente Mensch".

HEP

Human Error Probablitity. Zentrale Maßzahl der *HRA*. Auf das "Teilsystem Mensch" bezogene aufgabenabhängige *Versagenswahrscheinlichkeit*.

Irrtum

Auf indadäquates Wissen zurückführbarer *Fehler*$_4$. Typische und weit verbreitete - also überindividuelle - Irrtümer gehen auf *Denkfallen* zurück.

Komplexe Systeme

Systeme mit vielen Komponenten (Subsystemen), die miteinander eng verkoppelt sind. Speziell: Systeme aus Hard- und Software.

Korrektheit

Erfüllung der *Spezifikation R*. Übereinstimmung zwischen realisierter und spezifizierter *Funktion*$_1$. Funktionsfähigkeit.

Ein System mit der Funktion S ist genau dann korrekt, wenn es für jedes $x \in \text{dom}(R)$ einen Ausgabewert y liefert, und wenn für alle $(x, y) \in S$ mit $x \in \text{dom}(R)$ gilt, dass $(x, y) \in R$.

Korrektheitsbeweis

Verifikation.

Korrektheitswahrscheinlichkeit k

Wahrscheinlichkeit des Satzes "Für alle zulässigen Beanspruchungen (Eingaben) liefert das System richtige Ergebnisse (Ausgaben)".

Lebensdauer

Die Zeitspanne vom Anwendungsbeginn bis zum *Ausfall*.

Produkthaftung

Ersatzpflicht des Herstellers für Personen- und Sachschäden, die Verbrauchern durch fehlerhafte Produkte entstehen (Gesetz vom 1.1.1990). Hersteller müssen unabhängig von eigener Schuld haften, sofern ihre Produkte fehlerhaft und die Fehler nach dem Stand von Wissenschaft und Technik erkennbar waren (Gefährdungshaftung).

Programmierfehler

Fehler$_3$, der bei der Umsetzung der *Spezifikation* in das Programm entsteht.

Qualität

Die Gesamtheit der Merkmale und Merkmalswerte einer Einheit (Produkt, Dienstleistung, Tätigkeit, Organisation) bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen (DIN 55 350). Qualitätsmerkmale von Produkten sind: *Funktionsfähigkeit*, Brauchbarkeit (Bedienbarkeit), *Zuverlässigkeit*, *Verfügbarkeit*, Instandhaltbarkeit, *Sicherheit*, Umweltverträglichkeit, Wirtschaftlichkeit und Schönheit. Ein wichtiger Maßstab der Qualität ist die Kundenzufriedenheit.

Qualitätsmanagement (QM)

Management zur *Qualität*ssicherung. Unternehmen des produzierenden Gewerbes können ihr QM-System nach DIN EN ISO 9000 (1987) zertifizieren lassen. Andere Bereiche ziehen nach. Wird auch für den Hochschulbereich diskutiert.

Qualitätsmanagement, umfassendes (Total Quality Management, TQM)

Eine Managementmethode, die Qualität in den Mittelpunkt einer Organisation stellt (DIN EN ISO 8402, 1995). Wichtige Ziele: Kundenzufriedenheit, langfristiger Geschäftserfolg, Zufriedenheit des Personals auf allen Ebenen der Organisation, Nutzen für die Gesellschaft (in dieser Rangfolge). Siehe auch. *Qualitätssicherung* und *Qualitätsmanagement*.

Qualitätssicherung

Alle geplanten und systematischen Tätigkeiten innerhalb des QM-Systems, um ein ausreichendes zu Vertrauen schaffen, dass eine Einheit die *Qualität*sforderung erfüllen wird.

QM

Qualitätsmanagement.

Redundanz

Funktionsbereites Vorhandensein von mehr als für die vorgesehene *Funktion*$_1$ notwendigen technischen Mitteln.

Risiko, objektives

Schadenserwartungswert.

Risiko, subjektives

Erwartungswert des subjektiven Schadens.

Robustheit

Fähigkeit eines Objekts, auch bei Verletzung der spezifizierten Randbedingungen vereinbarte Funktionen zu erfüllen bzw. seine Funktionsfähigkeit zu erhalten (VDI-GIS, 1993). Der Begriff ist unscharf. Es bleibt offen, woran Robustheit zu messen ist, denn: *spezifizierte* Robustheit ist ein Widerspruch in sich.

Schnitzer

Fehler$_4$ auf der Ebene des automatisierten (routinierten) Denkens und Handelns.

Sicherheit

Abwesenheit von Gefahr (DIN VDE 31 000/2). Wird durch die Erfüllung sicherheitsbezogener Korrektheits- und Zuverlässigkeitsforderungen angestrebt bzw. erreicht (VDI 3542/1). Siehe auch *Sicherheitsspezifikation*, *Gefahr*$_2$.

Sicherheitsbezogene Korrektheit

Korrektheit, bezogen auf die *Sicherheitsspezifikation*.

Sicherheitsbezogene Korrektheitswahrscheinlichkeit

Korrektheitswahrscheinlichkeit, bezogen auf die *Sicherheitsspezifikation*.

Sicherheitsbezogene Spezifikation

Sicherheitsspezifikation.

Sicherheitsbezogene Versagenswahrscheinlichkeit

Versagenswahrscheinlichkeit, bezogen auf die *Sicherheitsspezifikation*.

Sicherheitsbezogener Fehler oder Ausfall

Fehler$_2$ eines Systems bezogen auf die *Sicherheitsspezifikation*.

Sicherheitsspezifikation R'

Obermenge der *Spezifikation* R, d.h.: Die Sicherheitsspezifikation lässt wenigstens die Eingabedatensätze der Spezifikation und die zugehörigen Ergebnisdatensätze zu. Die Sicherheitsspezifikation ist so zu gestalten, dass bei deren Einhaltung Gefahren$_1$ ausgeschlossen sind. Bei Verletzung der Sicherheitsspezifikation kann es zu sicherheitsbezogenen Fehlfunktionen kommen. Siehe auch: sicherheitsbezogene Spezifikation.

Spezifikation R

Genaue (möglichst formale) Beschreibung eines Objekts im Sinne der Aufgabenstellung. Dadurch wird eine Ein-Ausgaberelation R definiert, das ist die Menge aller zulässigen Paare (x, y) von Eingabe- (x) und Ausgabewerten (y).

Siehe auch: *Korrektheit*, *funktionale Spezifikation*, *Sicherheitsspezifikation*.

Spezifikationsfehler

Abweichung zwischen den in den Anforderungen festgelegten Eigenschaften und der *Spezifikation*. Siehe *Validation*.

Statistische Wahrscheinlichkeit (Wahrscheinlichkeit im statistischen Sinn)

Auf den relativen Häufigkeiten basierender Wahrscheinlichkeitsbegriff.

Störung$_1$

(Vorübergehende) Beeinträchtigung einer *Funktion*$_1$. *Fehler*$_1$ aufgrund vorübergehender Einflüsse (elektromagnetische Beeinflussung, Strahlung und dergleichen)

Störung$_2$

Fehlende, fehlerhafte oder unvollständige Erfüllung einer geforderten *Funktion*$_1$. Etwa gleichbedeutend mit *Fehler*$_1$.

Test

Überprüfung, ob ein *(deterministisches) System* S auf sämtliche Daten einer Teilmenge T der Menge der Eingabedaten X korrekt antwortet. Der Test ist genau dann bestanden (negativ), wenn $R(x, S(x))$ für alle $x \in T$ gilt. Ein Test heißt vollständig, wenn $T = \mathrm{dom}(R)$. Ein deterministisches System ist *korrekt*, wenn es einen vollständigen Test besteht.

THERP

Technique for Human Error Rate Prediction. Bekannte *HRA*-Methode.

TQM

Siehe unter: *Qualitätsmanagement, umfassendes*.

Überlebenswahrscheinlichkeit

Zuverlässigkeitsfunktion $Z(t)$ für den Fall, dass anstelle des Ereignisses *Versagen* der *Ausfall* gewählt wird. Siehe auch Zuverlässigkeitsfunktion, Ausfallrate, Versagensrate.

Unkorrektheitswahrscheinlichkeit u

Komplement zu eins der *Korrektheitswahrscheinlichkeit*: $u = 1 - k$. Auch: Fehlerwahrscheinlichkeit.

Validation (eines Produkts)

Der Nachweis, dass ein System den *Anforderungen* genügt.

Validation der Spezifikation

Überprüfung, ob die *Spezifikation* den *Anforderungen* genügt. Wenn V die Relation einer geforderten Eigenschaft ist, dann ist nachzuweisen, dass dom(V) in dom(R) enthalten ist und dass für alle $(x, y) \in R$ mit $x \in$ dom(V) folgt, dass $(x, y) \in V$.

Verfügbarkeit

Wahrscheinlichkeit für korrekte *Funktion*$_1$ im Sinne der *Zuverlässigkeit*$_1$.

Verifikation

Nachweis, dass ein Objekt die *Spezifikation* erfüllt. Nachweis der *Korrektheit*; *Korrektheitsbeweis*.

Verlässlichkeit

Zuverlässigkeit$_2$.

Versagen

Verhalten, das nicht der Spezifikation entspricht. Auftreten eines *Fehlers*$_1$.

Versagensabstand, mittlerer ~

Mittlerer Abstand des *Versagens* (Software). Siehe *Überlebenswahrscheinlichkeit*.

Versagensrate $\lambda(t)$

Absolutbetrag der Ableitung des natürlichen Logarithmus der *Zuverlässigkeitsfunktion*: $\lambda(t) = - \mathrm{d} \ln(Z(t))/\mathrm{d}t$. In Spezialfällen gleich dem Kehrwert des mittleren Versagensabstands.

Versagenswahrscheinlichkeit, mittlere ~, p

Wahrscheinlichkeit dafür, dass ein System (bzw. ein Programm) S und ein Beanspruchungsfall (Eingabedatensatz) x vorliegen derart, dass das System sich nicht *spezifikationsgemäß* verhält (das Berechnungsresultat falsch ist). Grad der Korrektheit eines komplexen Systems oder Programms.

Versagenswahrscheinlichkeit, datenabhängige , $p(x)$

Versagenswahrscheinlichkeit für einen festen Eingabewert x.

Zeit bis zum Versagen

Tritt bei der *Verlässlichkeit*sbewertung an die Stelle der *Lebensdauer*.

Zuverlässigkeit$_1$

Fähigkeit eines Systems, für eine gegebene Zeit korrekt zu arbeiten. Dabei wird vorausgesetzt, dass das System zu Anwendungsbeginn korrekt ist und nur Ausfälle zur Unkorrektheit führen können. Kenngrößen: *Zuverlässigkeitsfunktion* und daraus abgeleitete.

Zuverlässigkeit$_2$

Grad des Vertrauens, das aufgrund geringer *Versagenswahrscheinlichkeit*, großer *Korrektheitswahrscheinlichkeit* oder geringer *Ausfall*- bzw. *Versagensrate* gerechtfertigt erscheint. Auch: Verlässlichkeit.

Zuverlässigkeitsfunktion $Z(t)$

Die Zuverlässigkeitsfunktion ist gleich der Wahrscheinlichkeit dafür, dass das System im Zeitintervall von 0 bis zur Zeit t nicht *versagen* wird.

$F(t) = 1-Z(t)$ ist die Verteilungsfunktion der Zeit bis zum ersten Versagen.

Bei exponentialverteilter Zeit ist $Z(t) = \exp(-t/T)$. Der Parameter T dieser Verteilung ist gleich der mittleren Zeit bis zum Versagen. Siehe auch: *Versagensrate*, *Überlebenswahrscheinlichkeit*, *Ausfallrate*, *Verfügbarkeit*.

Literaturverzeichnis

Anderson, J. R.: *Cognitive Psychology and Its Implications* (4th Edition). Freeman, New York, Oxford 1995 (Deutsche Ausgabe: *Kognitive Psychologie.* Spektrum der Wissenschaft Verlag, Heidelberg 1988)

Andrews, J. D.; Moss, T. R.: *Reliability and Risk Assessment.* Longman Group UK 1993

Balzert, H.: *Lehrbuch der Software-Technik. Software-Entwicklung.* Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford 1996

Balzert, H.: *Lehrbuch Grundlagen der Informatik.* Spektrum Akademischer Verlag, Heidelberg, Berlin 1999

Bechmann, G. (Hrsg.): *Risiko und Gesellschaft. Grundlagen und Ergebnisse der interdisziplinären Risikoforschung.* Westdeutscher Verlag, Opladen 1993

Bechmann, G.: *Risiko als Schlüsselkategorie der Gesellschaftstheorie.* In: Bechmann, 1993, S. 237-276

Bernstein, P. L.: *Against the Gods. The remarkable story of risk.* Wiley, New York 1996.

Birolini, A.: *Qualität und Zuverlässigkeit technischer Systeme.* 3. Auflage. Springer, Berlin, Heidelberg 1991

Birolini, A.: *Reliability Engineering. Theory and Practice.* 3rd edition. Springer, Berlin, Heidelberg 1999

BMFT - Bundesministerium für Forschung und Technologie (Hrsg.): *Deutsche Risikostudie Kernkraftwerke der Gesellschaft für Reaktorsicherheit (GRS).* Verlag TÜV Rheinland, Köln 1980

Bubb, H.: *Ist Automatisierung die richtige Antwort auf menschliches Versagen?* VDI-Berichte1336 (1997), 117-131

Butler, R. W.; Finelli, G. B.: *The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software.* IEEE Trans. on Software Engineering, 19 (January 1993) 1, 3-12

Carnap, R; Stegmüller, W.: *Induktive Logik und Wahrscheinlichkeit.* Springer, Wien 1959

Conference, SAFECOMP '98, Heidelberg, Germany, October 1998 (Herausgeber: W. Ehrenberger). Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg 1998 (S. 89-99)

Cube, F. v.: *Gefährliche Sicherheit. Die Verhaltensbiologie des Risikos.* Hirzel, Stuttgart 1995

Dahl, O.-J.; Dijkstra, E. W.; Hoare, C. A. R.: *Structured Programming.* Academic Press, London 1972

Däßler, K.; Sommer, M.: *Pascal. Einführung in die Sprache, DIN-Norm 66256, Erläuterungen.* 2. Aufl. Springer-Verlag, Berlin, Heidelberg, New York 1985

Dearden, A. M.; Harrison, M. D.: *Impact as a Human Factor in Interactive System Design.* (In: Redmill, Anderson, 1996, S. 184-199)

Dearden, A. M.; Harrison, M. D.: *Risk Analysis, Impact and Interaction Modelling.* February 9, 1996 (private communication)

DGQ-Band 17-01: *Zuverlässigkeit komplexer Systeme aus Hardware und Software.* Beuth

Verlag, Berlin 1998

DGQ-ITG-Schrift Nr. 12-52: *Methoden und Verfahren der Software-Qualitätssicherung.* Frankfurt 1992

DIN EN ISO 8402, 1995: *Qualitäts-Begriffe*

DIN EN ISO 9000, 1987: *Qualitäts-Management- und Qualitätssicherungsnormen: Leitfaden zur Auswahl und Anwendung.*

DIN V 19251 *MSR-Schutzeinrichtungen, Anforderungen und Maßnahmen zur gesicherten Funktion*

DIN VDE 31 000, Teil 2 (Dezember 1987): *Allgemeine Leitsätze für das sicherheitsgerechte Gestalten technischer Erzeugnisse. Begriffe der Sicherheitstechnik. Grundbegriffe*

DIN 55 350, Teil 11, Mai 1987: *Begriffe der Qualitätssicherung und Statistik; Grundbegriffe der Qualitätssicherung*

DIN V 19 250 (Vornorm Januar 1989): *Messen-Steuern-Regeln. Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen*

Dörner, D., Schaub, H.: *Errors in Planning and Decision-making and the Nature of Human Information Processing.* Applied Psychology: An International Review 43 (1994) 4, 433-453

Douglas, M.: *How Institutions Think.* Syracuse University Press, Syracuse, New York 1986

Douglas, M.; Wildavsky, A.: *Risk and Culture.* University of California Press, Berkeley, Los Angeles 1982

Eckhardt, D. E.; Lee, L. D.: *A Theoretical Basis for the Analysis of Multiversion Software Subject to Coincident Errors.* IEEE Trans. on Software Engineering, SE-11 (1985) 12, 1511-1517

Ehrenberger, W.: *Software-Verifikation. Verfahren für den Zuverlässigkeitsnachweis von Software.* Hanser, München, Wien 2002

Eibl-Eibesfeldt, I.: *Die Biologie des menschlichen Verhaltens.* Piper, München 1984

Elzer, P. F.; Kluwe, R. H.; Boussoffara, B. (Eds): *Human Error and System Design and Management.* Lecture Notes in Control and Information Sciences 253. Springer, London 2000

Endsley, M. R.: *Toward a Theory of Situation Awareness in Dynamic Systems.* Human Factors 37 (1995) 1, 32-64

Evans, J. St. B. T.: *Bias in Human Reasoning: Causes and Consequences.* Lawrence Erlbaum, Hove and London 1989

Feynman, R. P.: *Kümmert Sie, was andere Leute denken? Neue Abenteuer eines neugierigen Physikers.* Piper, München 1988 (Teil 2 des Buches: Mr. Feynman geht nach Washington, um die Challenger-Katastrophe zu untersuchen)

Fisz, M.: *Wahrscheinlichkeitsrechnung und mathematische Statistik.* DVW Berlin 1976

Fritzsche, A. F.: *Wie sicher leben wir? Risikobeurteilung und -bewältigung in unserer Gesellschaft.* TÜV Rheinland, Köln 1986

Geiger, W.: *Qualitätslehre.* Vieweg, Braunschweig, Wiesbaden 1998

Gigerenzer; Gerd: *Die Evolution des statistischen Denkens* (*The Evolution of Statistic Thin-*

king). Published in: Unterrichtswissenschaft, 32 (2004) 1 – frei im Internet
Goldstein, E. B.: *Wahrnehmungspsychologie.* Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford 1997
Görke, W.: *Handbuch der Informatik 2.1. Fehlertolerante Rechensysteme.* Oldenbourg, München, Wien 1989
Grams, T. (1981): *Sicherheit aus der Sicht der technischen Zuverlässigkeit.* VDI-Berichte Nr. 395, 1981
Grams, T. (1990[a]): *Denkfallen und Programmierfehler.* Springer, Berlin, Heidelberg 1990
Grams, T. (1990[b]): *Software-Zuverlässigkeit, gibt es das? (Software reliability, is there such a thing?)* it 32 (1990) 2, 125/134
Grams, T. (1993): *Täuschwörter im Software Engineering.* Informatik Spektrum 16 (1993) 3, 165-166
Grams, T. (1997): *Ein allgemeines Zuverlässigkeitsmodell mit Anwendungen.* Automatisierungstechnik at (1997) 8, 378-385
Grams, T. (1998[a]): *Bedienfehler und ihre Ursachen.*
Teil 1. Automatisierungstechnische Praxis atp 40 (1998) 3, S. 53-56
Teil 2. Automatisierungstechnische Praxis atp 40 (1998) 4, S. 55-60
Grams, T. (1998[b]): *Das Elend der Zuverlässigkeitswachstumsmodelle.* Informatik-Spektrum (1998) 5, 291-295
Grams, T. (2000): *Putting the Normative Decision Model into Practice.* In: Elzer/Kluwe/Boussoffara 2000, S. 99-107
Grams, T. (2006): *Denkfallen – Klug irren will gelernt sein.* Vortrag zur MIND AKADAMIE 2006 vom 5.-8. Oktober in Marburg. Manuskript. Erschienen in „Jenseits des Verstandes – Beiträge aus Philosophie, Psychologie, Wirtschaft und Praxis“ (Eine Publikation des MinD Hochschul Netzwerkes, herausgegeben von Martin Dresler, Tanja Gabriele Klein), Hirzel Verlag, Stuttgart 2007, S. 13-24
Grams, T., Angermann, D.: *Eine einfache Methode zur quantitativen Bewertung der Sicherheit.* Regelungstechnische Praxis 23(1981)3, 103-108
Gries, D.: *The Science of Programming.* Springer Heidelberg 1981
Hammond, J. S.; Keeney, R. L.; Raiffa, H.: *Smart Choices.* Harvard Business School Press, Boston, Mass. 1999
Hell, W.; Fiedler, K.; Gigerenzer, G. (Hrsg.): *Kognitive Täuschungen. Fehl-Leistungen und Mechanismen des Urteilens, Denkens und Erinnerns.* Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford 1993
Hendrick, H. W.: *Organizational Design and Macroergonomics.* In: Salvendy, 1997, S. 594 ff.
Hering, E.; Triemel, J.; Blank, H.-P. (Hrsg.): *Qualitätsmanagement für Ingenieure.* VDI Verlag, Düsseldorf 1996
Hohler, B.; Villinger, U.: *Normen und Richtlinien zur Qualitätssicherung von Steuerungssoftware.* Informatik-Spektrum 21 (1998) 2, 63-72
Hörstel, J.; Ritzau, H.-J.: *Fehler im System.* Ritzau KG - Verlag Zeit und Eisenbahn, Pürgen 2000

Humphrey, W. S.: *A Discipline of Software Engineering.* Addison-Wesley, Reading, Mass. 1995

Hütte: *Die Grundlagen der Ingenieurwissenschaften.* 25. Auflage. Springer, Berlin, Heidelberg 1991

Item Software: *FaultTree+ for Windows - Version 8.0. Fault and Event Tree Analysis Program.* Item Software (UK) 1998

Kahneman, D.; Slovic, P.; Tversky, A. (eds.): *Judgment under uncertainty: Heuristics and biases.* Cambridge University Press, Cambridge, 1982

Kahneman, D.; Tversky, A.: *Prospect Theory: An Analysis of Decision under Risk.* Econometrica, 47 (March, 1979) 2, 263-291

Kahneman, D.; Tversky, A.: *The simulation heuristic.* (In: Kahneman, Slovic, Tversky, 1982, S. 201-208)

Kaiser, R.: *Object Pascal mit Delphi.* Springer, Berlin, Heidelberg 1997

Kernighan, B. W.; Ritchie, D. M.: *The C Programming Language.* Prentice Hall, Englewood Cliffs, N. J. 1988

Kleinrock, L.: *Queueing Systems.* Volume 1: Theory. Wiley, New York 1975

Knight, J. C.; Leveson, N. G.: *Correlated failures in multiversion software.* SAFECOMP '85. Proceedings of the 4th IFAC Workshop, Como, Italy, 1985 (Pergamon Press), 159-165

Knuth, D.: *The Art of Computer Programming. Vol. 1: Fundamental Algorithms.* Addison-Wesley 1973

Kozlow, B. A.; Uschakov, I. A.: *Reliability Handbook.* Holt, Rinehart & Winston, New York 1970 (Deutsch: Koslow, B. A.; Uschakow, I. A.: *Handbuch zur Berechnung der Zuverlässigkeit für Ingenieure.* Hanser, München 1979)

Krech, Crutchfield u. a.: *Grundlagen der Psychologie.* 7 Bände. Studienausgabe. Beltz, Weinheim 1992

Kuhlmann, A.: *Einführung in die Sicherheitswissenschaft.* Vieweg, Wiesbaden, 1981/TÜV Rheinland, Köln 1981

Kuhn, T. S.: *Die Struktur wissenschaftlicher Revolutionen.* 2. revidierte Auflage. Suhrkamp, Frankfurt/M. 1976

Kuhn, T. S.: *The Structure of Scientific Revolutions.* 3. ed. The University of Chicago Press 1996

Leveson, N. G.: *Safeware. System Safety and Computers. A Guide to Preventing Accidents and Losses caused by Technology.* Addison-Wesley, Reading, Massachusetts 1995

Lewis, H. W.: *Die Sicherheit von Kernkraftwerken.* Spektrum der Wissenschaft (1980) 5, 30-43

Linger, R. C.; Mills, H. D.; Witt, B. I.: *Structured Programming. Theory and Practice.* Addison-Wesley, Reading, Mass. 1979

Lions, J. L.: *ARIANE 5. Flight 501 Failure.* Report by the Inquiry Borad. The Chairman of the Board. Paris 19.7.1996

Littlewood, B.; Miller, D. R.: *A conceptual model of the effect of diverse methodologies on coincident failures in multi-version software.* Informatik-Fachberichte 147 (Herausge-

ber: F. Belli und W. Görke), S. 263-272 , Springer-Verlag, Berlin, Heidelberg 1987
Litz, L. (Hrsg.): *Sicherheitsgerichtete Automatisierungstechnik.* Schwerpunktheft Automatisierungstechnik 46 (1998) 2
Liu, Y.: *Software-user Interface Design.* In: Salvendy, 1997, S. 1689 ff.
Lorenz, K.: *Die Rückseite des Spiegels. Versuch einer Naturgeschichte menschlichen Erkennens.* Piper, München 1973
Lorenzen, P.: *Formale Logik.* De Gruyter, Berlin 1970
Luchins, A. S.: *Mechanization and Problem Solving. The Effect of Einstellung.* Psychological Monographs 54 (1942)
Luhmann, N.: *Die Moral des Risikos und das Risiko der Moral.* In: Bechmann, 1993, S. 327-338
Lyu, M. R. (edt.): *Handbook of Software Reliability Engineering.* McGraw-Hill, New York 1996
Medwedjew, Z.: *Das Vermächtnis von Tschernobyl.* Daedalus, Münster 1991
Mellor, P.: *CAD: Computer-Aided Disasters.* High Integrity Systems 1 (1994) 2
Mérö, L.: *Optimal entschieden? Spieltheorie und die Logik unseres Handelns.* Birkhäuser, Basel 1998
Meyer, B.: *Object-oriented Software Construction.* Prentice Hall, New York, London, Toronto, Sydney, Tokyo 1988
MIL-HDBK-217: *Military Handbook Reliability Prediction of Electronic Equipment.* National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161, Part 217-D: Jan. 1982; Part 217-E: Oct. 1986.
Mili, A.: *An Introduction to Program Fault Tolerance. A Structured Programming Approach.* Prentice Hall, Englewood Cliffs, NJ 1990
Mills, H. D.: *Structured Programming: Retrospect and Prospect.* IEEE Software (November 1986), 58-66
Miranda, E.: *The use of reliability growth models in project management.* ISSRE '98, 291-298
Myers, G. J.: *Methodisches Testen von Programmen.* Oldenbourg Verlag, München 1987
Myers, G. J.: *Software Reliability. Principles and Practices.* Wiley, New York 1976
Neumann, P. G.: *Computer Related Risks.* The ACM Press1995
Norman, D. A.: *Categorization of Action Slips.* Psychological Review 88 (1981) 1, 1-15
Norman, D. A.: *The Design of Everyday Things.* Double Day Currency, New York 1988
O´Connor, P. D. T.: *Practical Reliability Engineering.* John Wiley & Sons 1991
Perrow, C.: *Normal Accidents.* Living with High-Risk Technologies. Princeton University Press 1999
Perrow, C.: *Normale Katastrophen. Die unvermeidlichen Risiken der Großtechnik.* Campus, Frankfurt/M., New York 1987
Pólya, Georg: *Mathematik und plausible Schließen. Band 2: Typen und Strukturen plausibler Folgerung.* Birkhäuser, Basel 1962
Popper, K. R.: *Objektive Erkenntnis. Ein evolutionärer Entwurf.* Hoffmann und Campe, Hamburg 1973

Popper, K.: *Logik der Forschung.* Mohr, Tübingen 1982

Poundstone, W.: *Labyrinths of Reason.* Anchor Book, New York 1988

Preparata, F. P.; Metze, G.; Chien, R. T.: *On the Connection Assignment Problem of Diagnosable Systems.* IEEE Trans. EC-16 (1967) 3, 848-854

Rasmussen, J.: *The Definition of Human Error and a Taxonomy for Technical System Design.* In: Rasmussen, J. et al. (edt.): New Technology and Human Error. Wesley, New York 1987

Rasmussen, J.; Duncan, K.; Leplat, J.: *New Technology And Human Error.* Wiley 1987

Reason, J.: *Human Error.* Cambridge University Press 1990

Reason, J.: *Menschliches Versagen.* Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford 1994)

Redmill, F.; Anderson, T. (Eds.): *Safety-critical Systems: The Convergence of High Tech and Human Factors.* Proc. of the 4th Safety-critical Systems Symposium, Leeds 1996. Springer, London 1996

Reiser, M.; Wirth, N.: *Programming in Oberon. Steps beyond Pascal and Modula.* ACM Press, New York 1992

Renn, O.; Zwick, M. M.: *Risiko- und Technikakzeptanz.* Springer, Berlin, Heidelberg 1997

Riedl, R.: *Biologie der Erkenntnis.* Paul Parey, Berlin, Hamburg 1981

Sachs, L.: *Angewandte Statistik. Anwendung statistischer Methoden.* Springer, Berlin, Heidelberg 1992

Sacks, O.: *Der Mann, der seine Frau mit einem Hut verwechselte.* Rowohlt, 1987

Saglietti, F.; Ehrenberger, W.; Kersken, M.: *Software-Diversität für Steuerungen mit Sicherheitsverantwortung.* Schriftenreihe der Bundesanstalt für Arbeitsschutz - Forschung - Fb 664, Dortmund 1992

Salvendy, G. (Edt.): *Handbook of Human Factors and Ergonomics* (2nd edition). John Wiley, New York 1997

Schneeweiss, W. G.: *Boolean Functions - with Engineering Applications and Computer Programs.* Springer, Berlin, Heidelberg, New York 1989

Sedmak, R. M.; Liebergot, H. L.: *Fault-tolerance of general purpose computer implemented by very large scale integration.* FTCS-8 (1978), 137-143.

Sharit, J.: *Allocation of Functions.* In: Salvendy, 1997, S. 301 ff.

Sheridan, T. B.: *HCI in Subervisory Control: Twelve Dilemmas.* In: Elzer/Kluwe/Boussoffara 2000, S. 1-12

Sheridan, T. B.: *Supervisory Control.* In: Salvendy, 1997, S. 1199 ff.

Shneiderman, B.: *Designing the User Interface.* Addison-Wesley, Reading, Mass. 1998

Shooman, M. L: *Probabilistic Reliablilty: an Engineering Approach.* Robert E. Krieger Publishing Company, Malabar, Florida 1990

Singer, W.: *Hirnentwicklung und Umwelt.* Spektrum der Wissenschaft (1985) 3, 48-61

Spektr. d. Wiss 1/1997: *Entwicklung & Technologie. Qualitäts-Management.* Spektrum der Wissenschaft (Januar 1997) 1, 96-105

Starr, C.: *Social Benefits vs. Technological Risk, What is our society willing to pay for safety?*

Science, 165 (1969) 19, 1232-1238. Deutsch: *Sozialer Nutzen versus technisches Risiko.* Erschienen in Bechmann (1997) S. 3-24

Stroustrup, B.: *The C++ Programming Language.* 2nd ed. Addison-Wesley, Reading, Mass. 1991

Székely, G. J.: *Paradoxa. Klassische und neue Überraschungen aus Wahrscheinlichkeitsrechnung und mathematischer Statistik.* Harri Deutsch, Thun, Frankfurt/M. 1990

Trimpop, R.: *Safety Culture.* In: Elzer/Kluwe/Boussoffara 2000, S. 189-199

Trompeta, B.; Wettingfeld, K.: *Auslegung von PLT-Schutzeinrichtungen entsprechend der Risikobetrachtungen in petrochemischen Anlagen.* Automatisierungstechnische Praxis atp 38 (1996) 10, 9-18

Tversky, A.; Kahneman, D.: *Evidential impact of base rates.* (In: Kahneman, Slovic, Tversky, 1982)

Tversky, A.; Kahnemann, D.: *Judgment under Uncertainty: Heuristics and Biases.* Science 185 (Sept. 1974), 1124-1131. (Reprint: Kahneman, Slovic, Tversky, 1982)

Vaughan, D.: *The Challenger Launch Decision.* The University of Chicago Press 1996

VDE/VDI-AG: *Sicherheitskonzepte der Prozeßleittechnik in verfahrenstechnischen Anlagen und in Kraftwerken.* vde-verlag, Berlin, Offenbach 1987

VDI Berichte 1336: *Sicherheitstechnik und Automatisierung.* VDI/VDE-GMA-Tagung Langen, 10./11. April 1997. VDI-Verlag, Düsseldorf (1997)

VDI/VDE-Richtlinie 2180, Blatt 1 bis 4, 1996: *Sicherung von Anlagen der Verfahrenstechnik mit Mitteln der Prozeßleittechnik*

VDI/VDE-Richtlinie 3541: *Steuerungseinrichtungen mit vereinbarter gesicherter Funktion*
Blatt 1 (Oktober 1985): *Einführung, Begriffe, Erklärungen*
Blatt 2 (Oktober 1985): *Vereinbarung der gesicherten Funktion*
Blatt 3 (Oktober 1985): *Maßnahmen für die Erstellung*
Blatt 4 (Juli 1995): *Maßnahmen im Betrieb*

VDI/VDE-Richtlinie 3542: *Sicherheitstechnische Begriffe für Automatisierungssysteme/Safety terms for automation systems*
Blatt 1 (Oktober 2000): *Qualitative Begriffe*/Part 1 (October 2000): *Qualitative terms and definitions*
Blatt 2 (Oktober 2000): *Quantitative Begriffe*/Part 2 (October 2000): *Quantitative terms and definitions*
Blatt 3 (Oktober 2000): *Anwendungshinweise und Beispiele*/Part 3 (October 2000): *Application information and examples*
Blatt 4 (Oktober 2000): *Zuverlässigkeit und Sicherheit komplexer Systeme (Begriffe)*/Part 4 (October 2000): *Reliability and safety of complex systems (terms)*

VDI-Berichte 1239: *Erfolg durch zuverlässige Technik.* Tagung Fulda, 26./27. September 1995. VDI-Verlag, Düsseldorf 1995

VDI-GIS (VDI-Gemeinschaftsausschuss Industrielle Systemtechnik, Herausgeber): *Software-Zuverlässigkeit - Grundlagen, konstruktive Maßnahmen, Nachweisverfahren.* VDI-Verlag, Düsseldorf 1993

VDI-Richtlinie 4001 Blatt 2, Juni 1986: *Begriffsbestimmungen zum Gebrauch des VDI-Zuverlässigkeitshandbuchs*

VDI-Richtlinie 4004, Blatt 2 (August 1986): *Zuverlässigkeitskenngrößen, Überlebenskenngrößen*

Weidlich, S.: *Aktualisierung der VDI/VDE-Richtlinie 2180 "Sicherung von Anlagen der Verfahrenstechnik".* In: VDI Berichte 1336 (1997)

Wirth, N.: *Gedanken zur Software-Explosion.* Informatik-Spektrum, 1994, Heft 1, S. 5-10

Sachverzeichnis

1

2

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W



Z